

MSc Computer Engineering  
Software Systems Engineering

# Mobility Behavior Monitoring

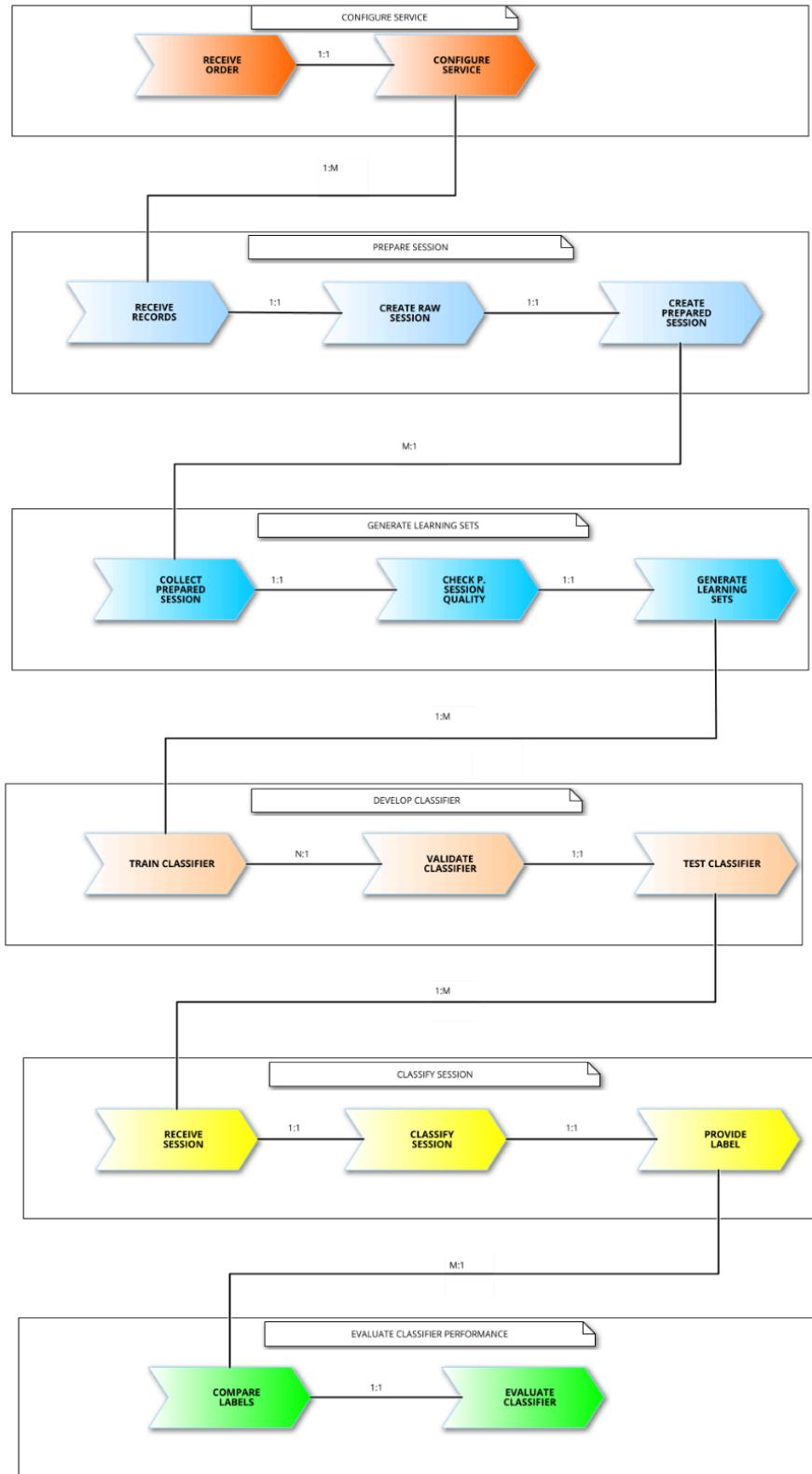
Lorenzo Bataloni  
Arsalen Bidani  
Claudio Daka  
Edoardo Pantè

Academic Year 2023/24

<b>Information System</b>	<b>3</b>
<b>BPMN Modeling</b>	<b>4</b>
<b>Use Cases</b>	<b>8</b>
<b>Sequence Diagrams</b>	<b>14</b>
<b>Class Diagrams (Analysis)</b>	<b>21</b>
<b>Mockups</b>	<b>26</b>
<b>Class Diagrams (Design)</b>	<b>29</b>
<b>Testing (Non-Resiliency)</b>	<b>33</b>
<b>Testing (Non-Automation)</b>	<b>37</b>
<b>Testing (Non-Interoperability)</b>	<b>44</b>
<b>Testing (Non-Responsiveness)</b>	<b>45</b>

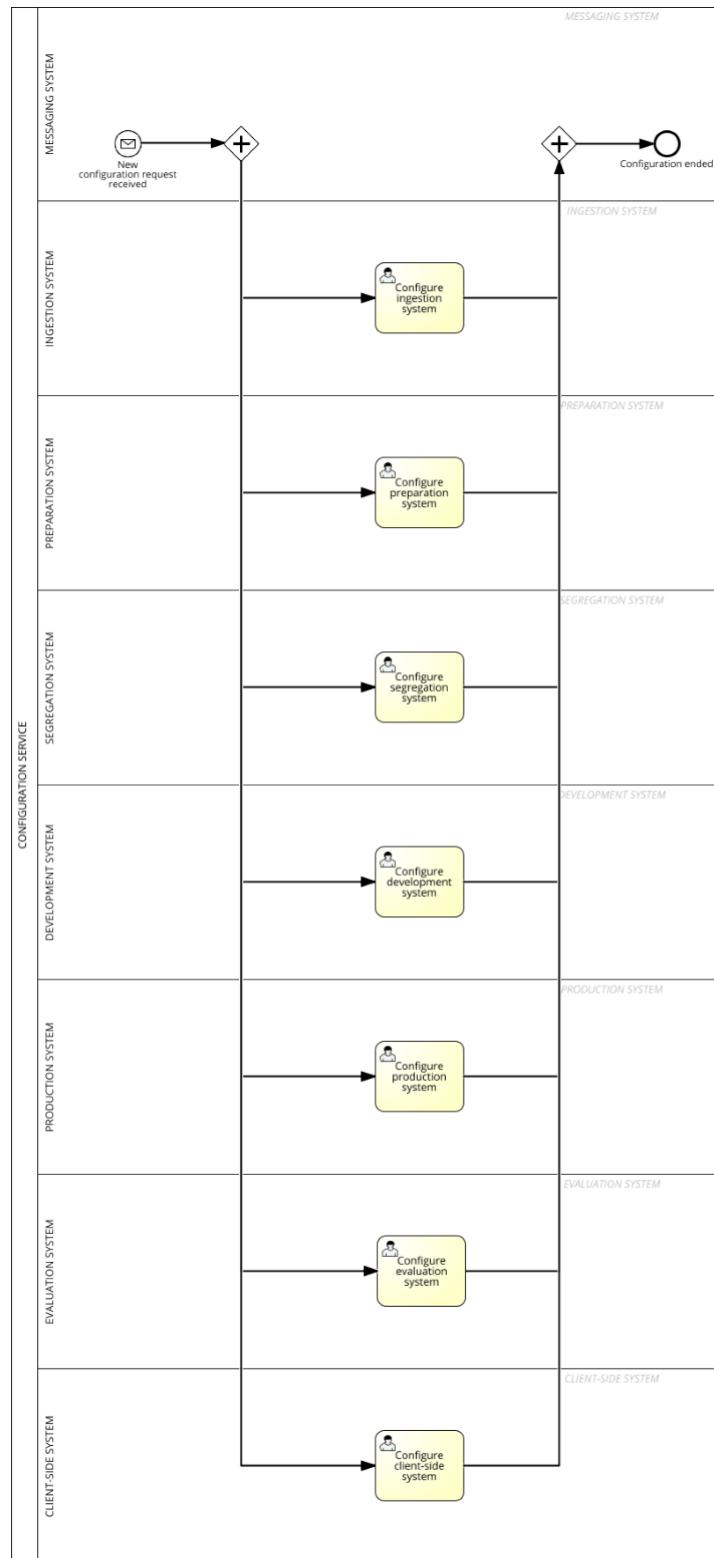
# Information System

## Process Landscape (Pantè, Bidani)

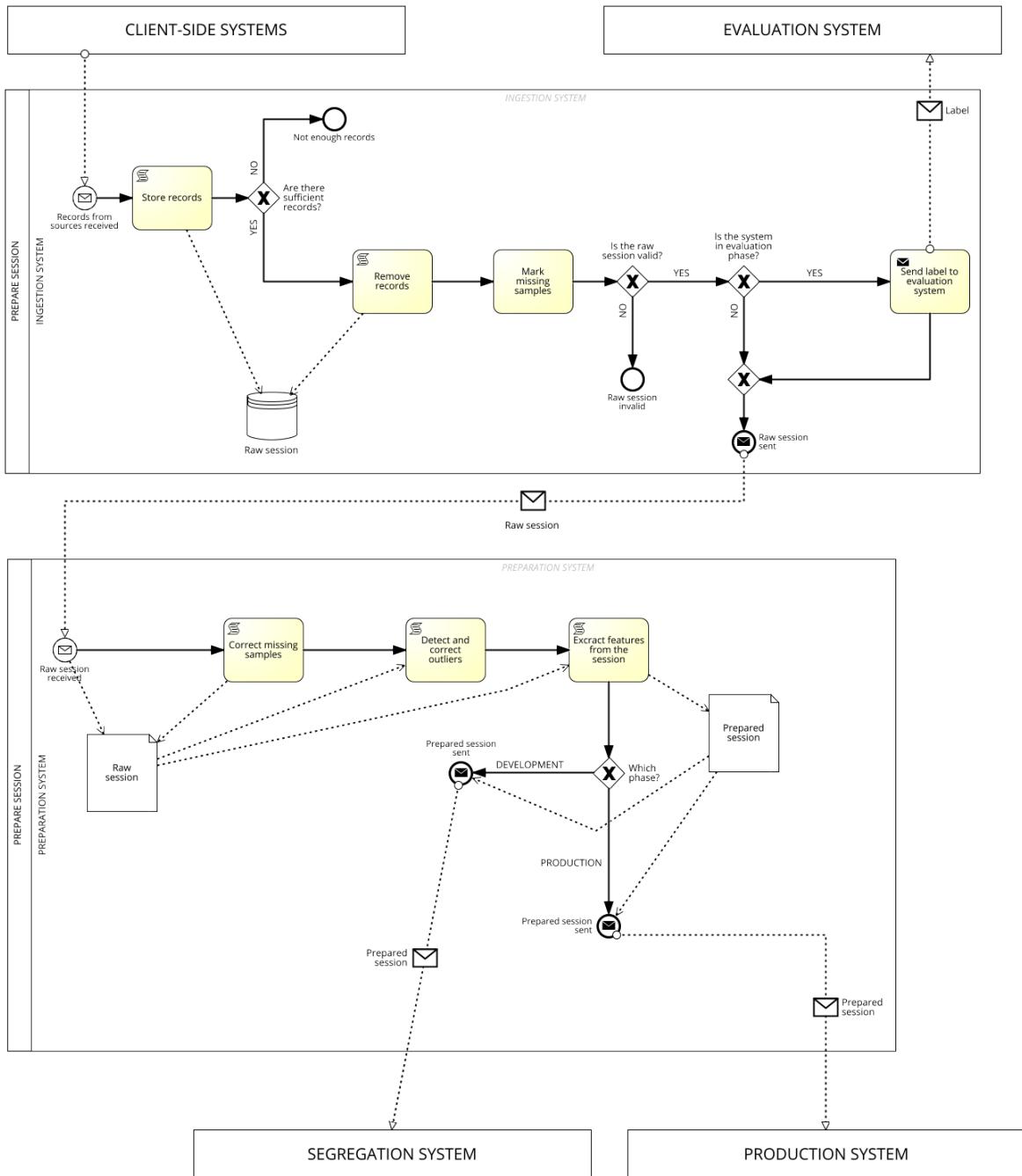


# BPMN Modeling

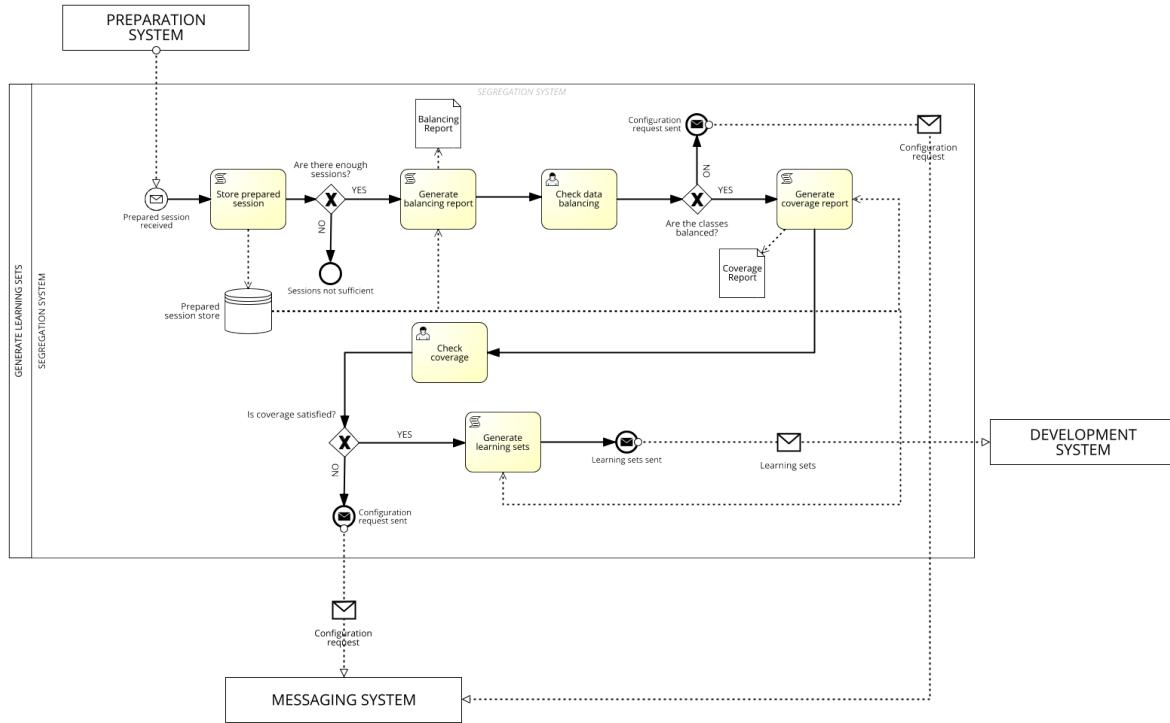
## 1 - CONFIGURATION PROCESS (Daka)



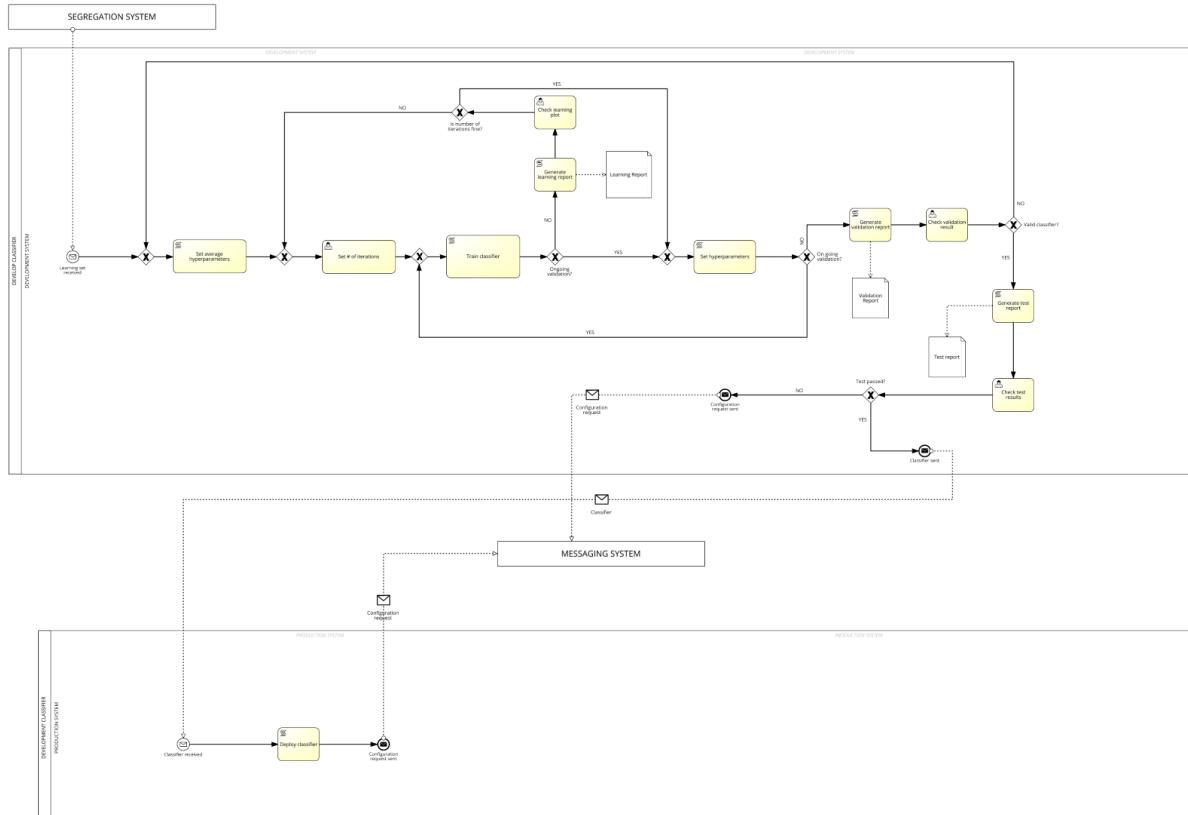
## 2 - PREPARE SESSION PROCESS (Daka)



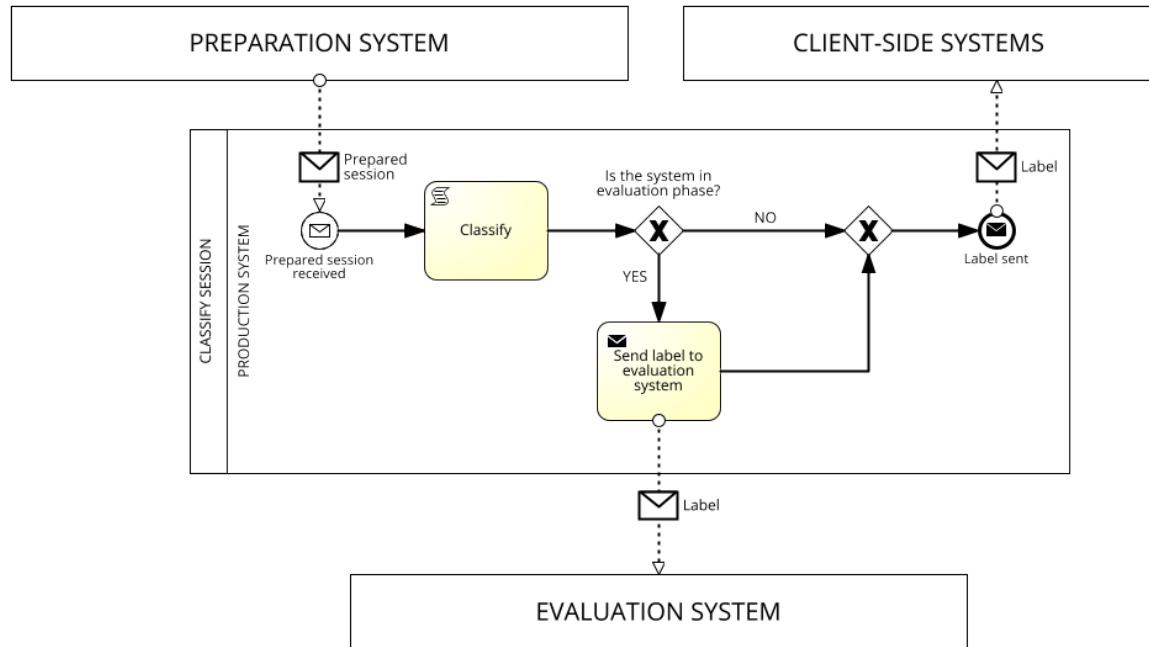
### 3 - GENERATE LEARNING SETS PROCESS (Pantè)



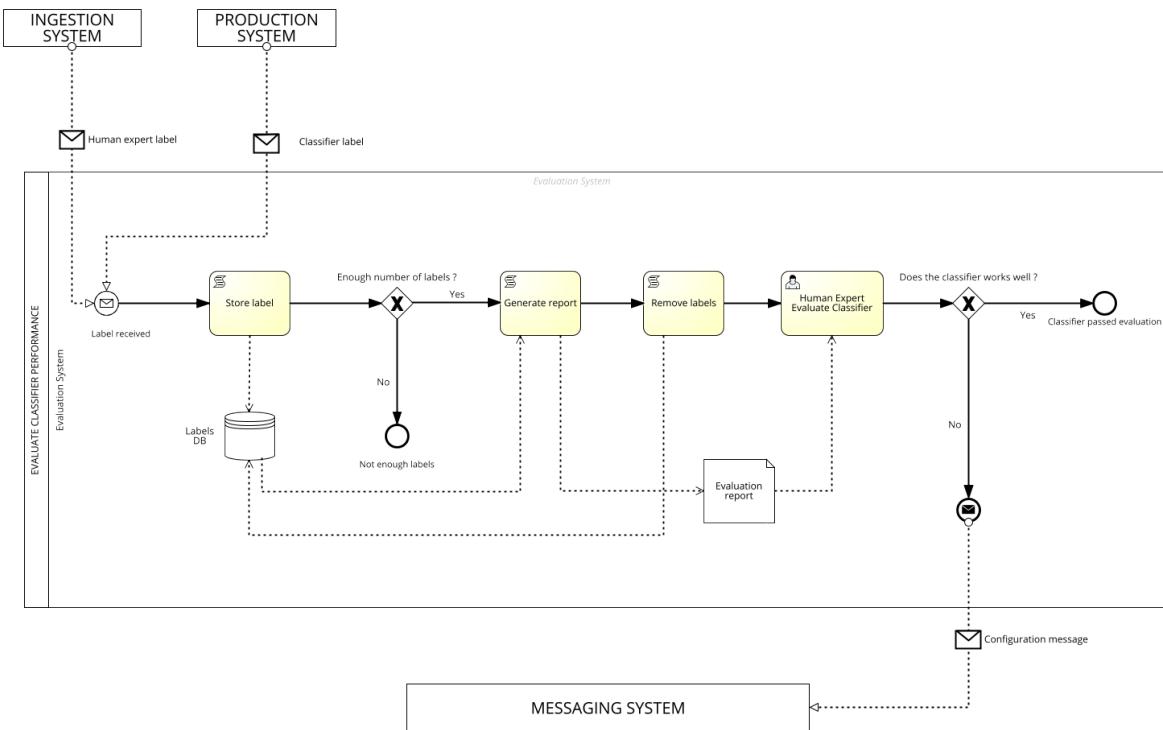
### 4 - DEVELOP CLASSIFIER (Bataloni)



## 5 - CLASSIFY SESSION (Pereira, Daka)

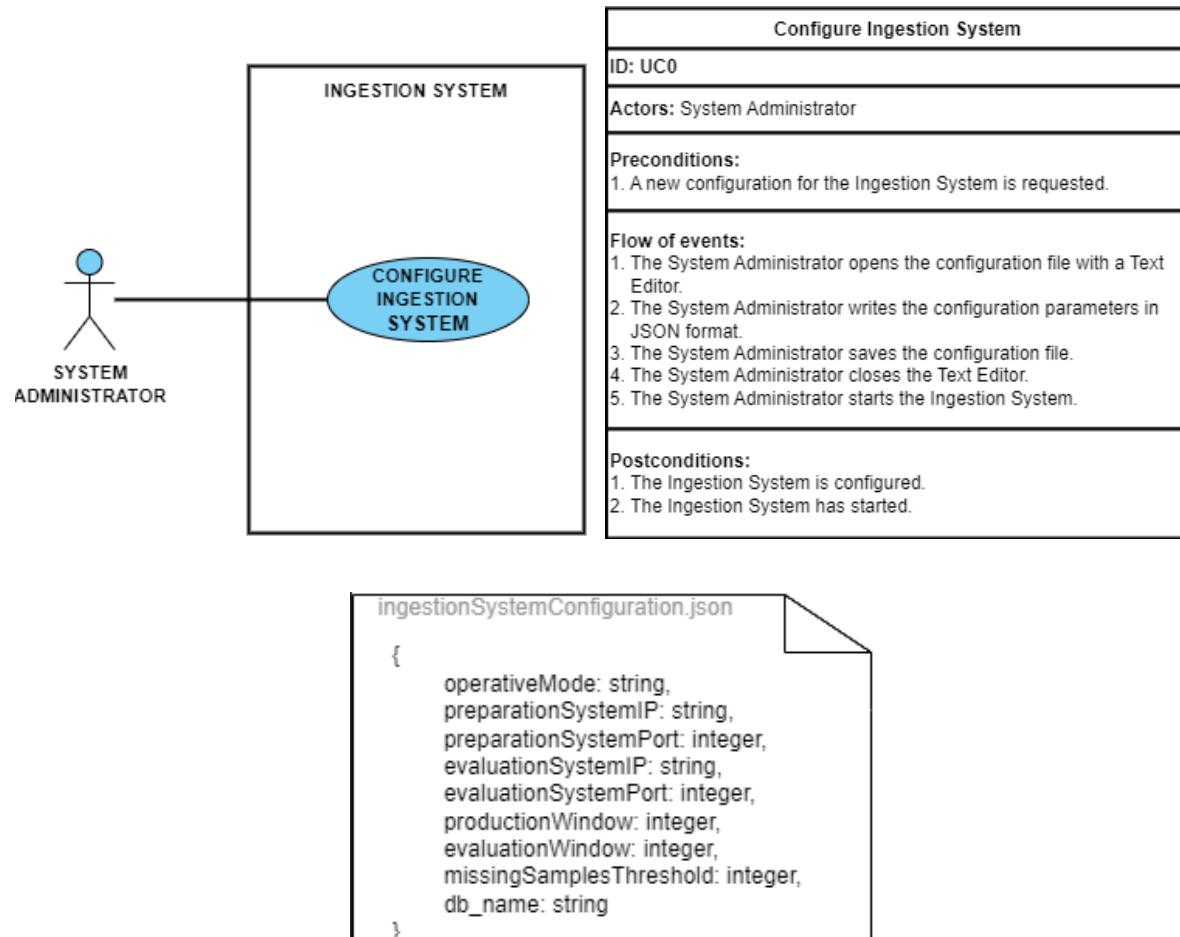


## 6 - EVALUATE CLASSIFIER PERFORMANCE (Bidani)

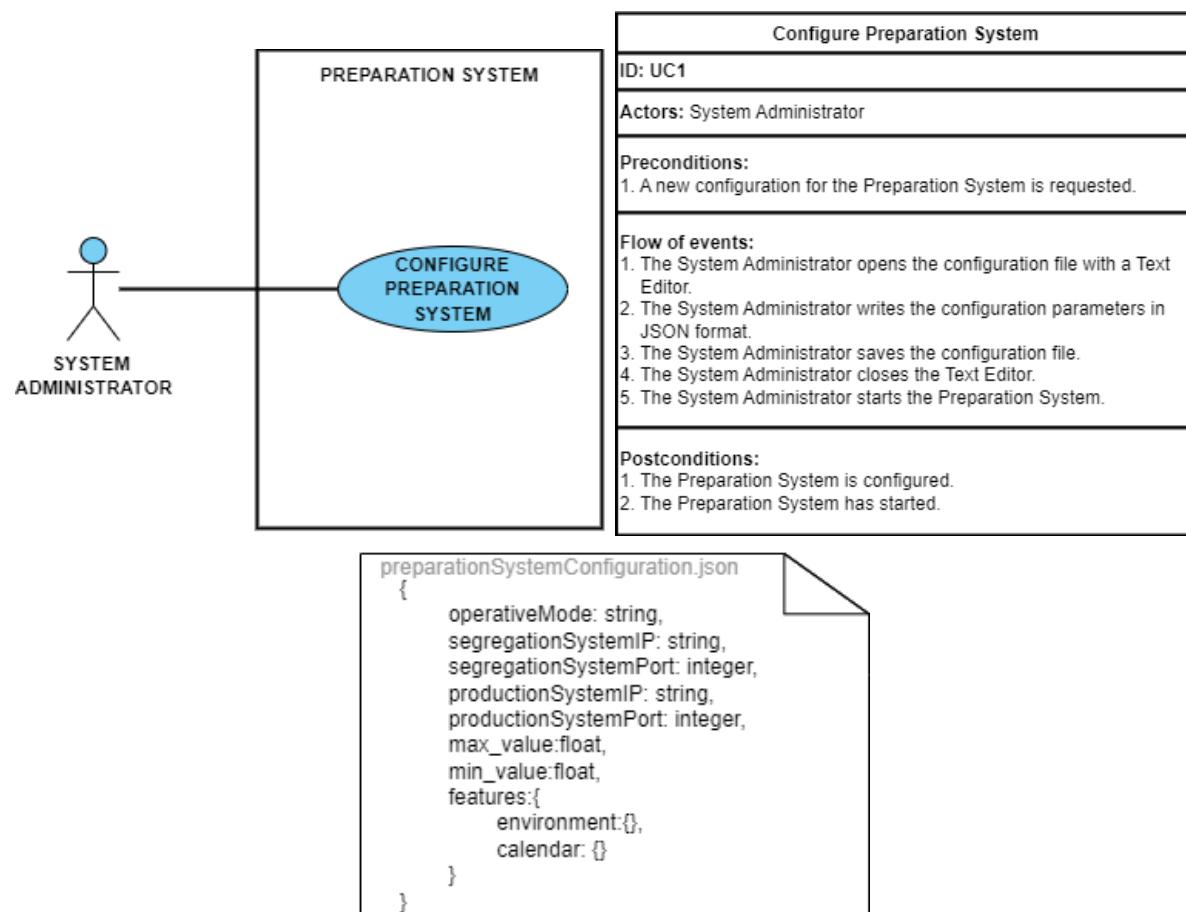


# Use Cases

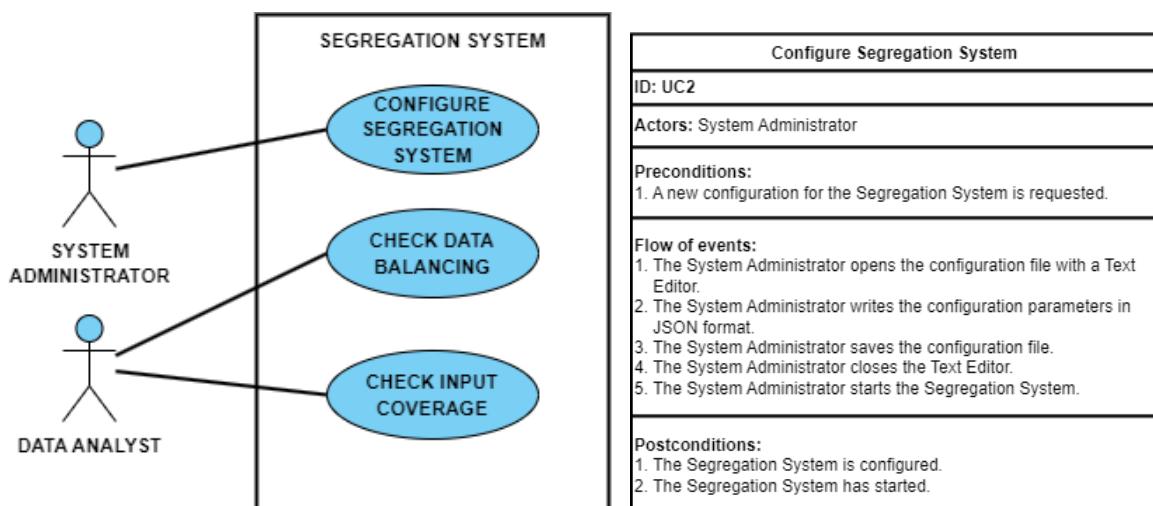
## Ingestion System (Daka)



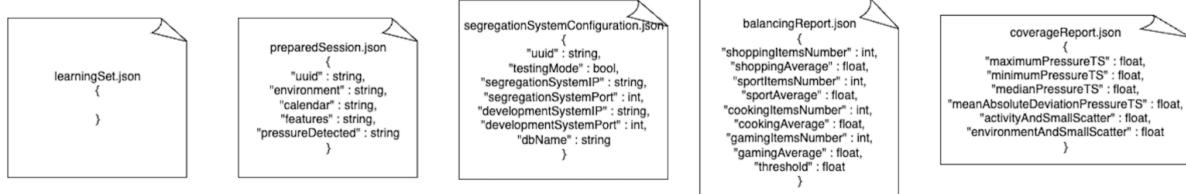
## Preparation System (Daka)



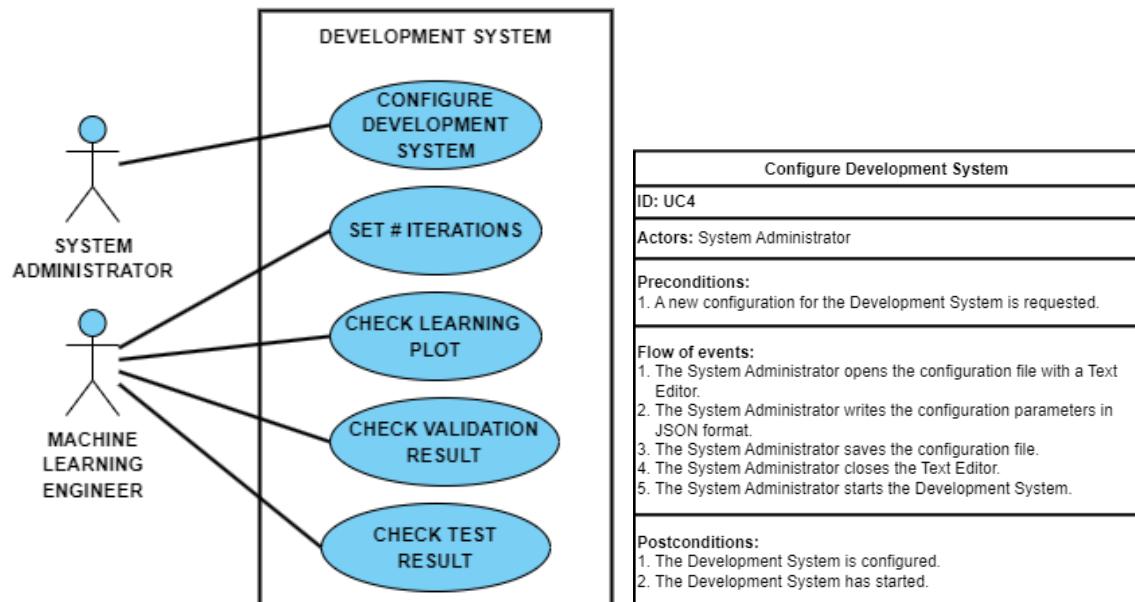
## Segregation System (Pantè, Bataloni, Daka)



Check Data Balancing	
ID: UC3	
<b>Actors:</b> Data Analyst	
<b>Preconditions:</b>	
1. The application has produced different plots to represent the labels distributions. 2. The application has stopped its execution waiting for a user input.	
<b>Flow:</b>	
1. The Data Analyst launches the plot viewer 2. The Data Analyst checks whether the Shopping samples are well balanced 3. The Data Analyst checks whether the Sport samples are well balanced 4. The Data Analyst checks whether the Cooking are well balanced 5. The Data Analyst checks whether the Gaming samples are well balanced 6. If the samples are well balanced then 6.1 The Data Analyst enters "Balanced" 7. Else 7.1 The Data Analyst fill the input form with the missing number of samples for each label 7.2 The Data Analyst enter "Send" and a new request is sent	
<b>Postconditions:</b>	
<b>Alternative flow 1:</b>	
1. The system continue its execution	
<b>Alternative flow 2:</b>	
1. A new configuration is requested	
<b>Check Input Coverage</b>	
ID: UC07	
<b>Actors:</b> Data Analyst	
<b>Preconditions:</b>	
• Application has generated as result a report about balancing. • Application has stopped waiting for user to press OK or NOT OK button.	
<b>Flow of events:</b>	
1. Data Analyst open the application with the balancing report. 2. Data Analyst analyze the report to check if the input space is correctly covered. 3. <b>FOR EACH</b> parameters, the Data Analyst: 1. IF the distribution is not uniform. 1. Data Analyst click the NOT OK button. 2. Data Analyst request a new Data Configuration for input. 3. Data Analyst close the balancing report window. 4. User correctly checked all the parameters. 5. Data Analyst press the OK button.	
<b>Postconditions:</b>	
If the Data Analyst press OK button, then learning set is generator, otherwise is necessary to check again the input coverage with other sets.	

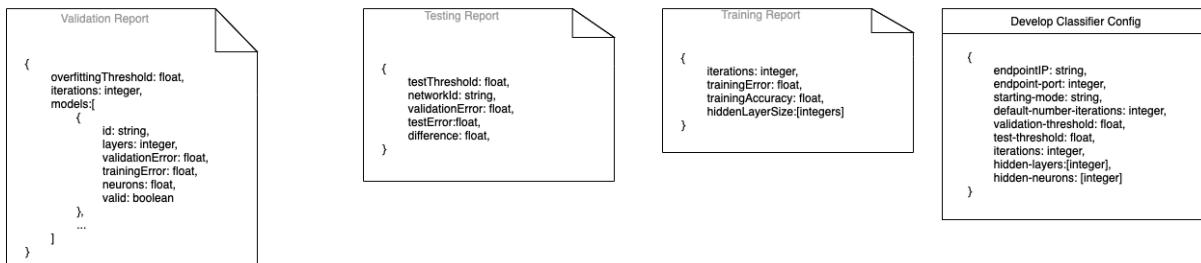


## Development System (Bataloni, Pantè, Bidani, Daka)

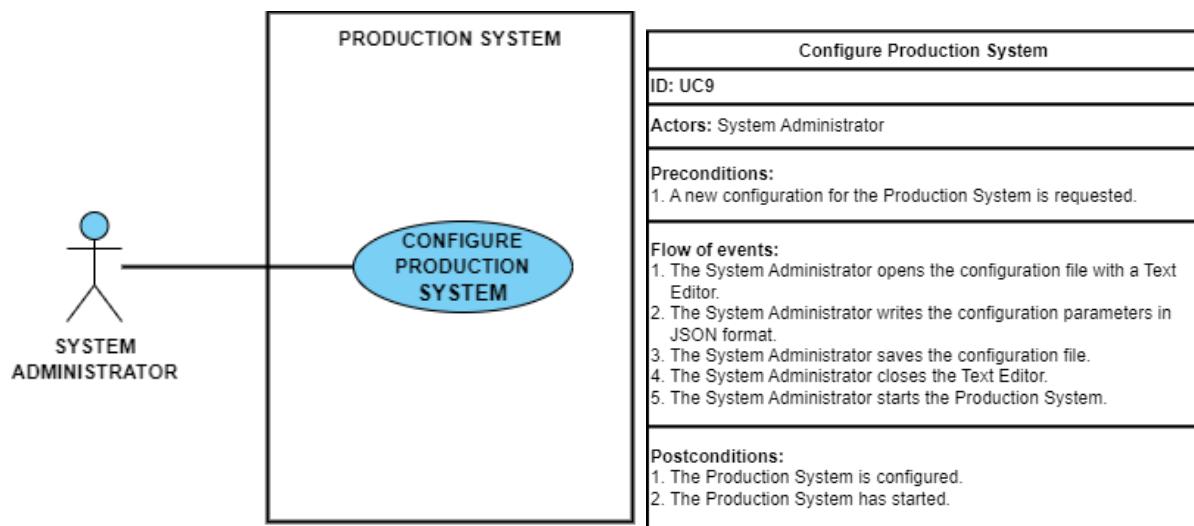


Set # Of Iterations
<b>ID:</b> UC05
<b>Actors:</b> Machine Learning Engineer
<b>Preconditions:</b>
<b>Flow of events:</b>
<p>1. Machine Learning Engineer open the application.      2. Machine Learning Engineer insert the number of iterations.      3. Machine Learning Engineer confirms by clicking the 'OK' button.</p>
<b>Postconditions:</b>
The system has setted the number of iterations and start the learning process with the specified number of iterations.
Check Learning Plot
<b>ID:</b> UC6
<b>Actors:</b> Machine Learning Engineer
<b>Preconditions:</b>
<p>1. An early training phase is completed      2. The training report has been computed</p>
<b>Flow:</b>
<p>1. The Machine Learning Engineer open the training plot.      2. If the curve slope is not too stiff:          2.1 The Machine Learning Engineer press OK      3. Else:          3.1 The Machine Learning Engineer send a new configuration with the number of iterations</p>
<b>Postconditions:</b>
1. The system continue its execution

Check Validation Result
<b>ID:</b> UC7
<b>Actors:</b> Machine Learning Engineer
<b>Preconditions:</b>
<p>1. The validation has been completed.      2. The report of the top 5 classifiers has been generated.</p>
<b>Flow of events:</b>
<p>1. The Machine Learning Engineer opens the file containing the report of the classifiers.      2. For each classifier in the report          2.1. The Machine Learning Engineer compares the validation error against the training error.          2.2. If the difference is greater than the given tolerance threshold              2.2.1. The classifier is discarded.          end for each.      3. If there is not any valid classifier left          3.1. The Machine Learning Engineer clicks on "None".      4. Else          4.1. If there is only one valid classifier left              4.1.1. The Machine Learning Engineer chooses that classifier as the best.          4.2. Else if there are more than one classifier with very similar error              4.2.2. The Machine Learning Engineer chooses the classifier with the lower complexity based on the number of layers and the number of neurons per layer.</p>
<b>Postconditions:</b>
<b>Alternative flow 1:</b>
1. The best classifier has been selected.
<b>Postconditions:</b>
<b>Alternative flow 2:</b>
1. New average hyperparameters have been setted. 2. A new training has started.
Check Test Results
<b>ID:</b> AC02
<b>Actors:</b> Human Expert
<b>Preconditions:</b>
The development system has generated the test results report by choosing the winner classifier and showcasing error results.
<b>Flow:</b>
<p>1. The Human Expert retrieves the test results report from the System.      2. The Human Expert opens the report.          2.1. Human Expert computes the difference between validation and test error.          2.2. Human Expert evaluates the difference with the tolerance cap.      3. If calculated difference &lt; tolerance cap.          3.1. Human Expert clicks on the "Accept network" button to adopt the chosen classifier.          3.2. System sends the classifier to the production system.      4. else          4.1. Human Expert clicks on "Refuse network" button to request a new configuration.          4.2. System requests a new classifier configuration          end if      5. The Human Expert closes the report.</p>
<b>Postconditions:</b>
A classifier network is chosen as a winner and consequently sent to the production system.

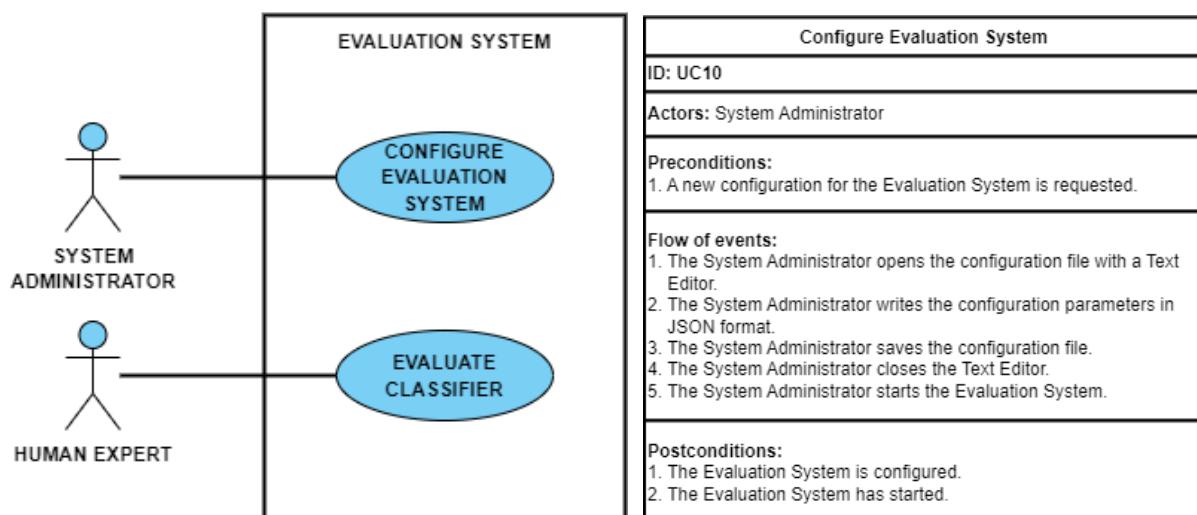


## Production System (Daka)



```
productionSystemConfiguration.json
{
  operativeMode: string,
  IP:string,
  port: integer,
  testing: boolean,
  thresholdValue: integer,
  evaluationWindow: integer,
  productionWindow: integer
}
```

## Evaluation System (Pantè, Daka, Bidani)



<b>Evaluate Classifier</b>
<b>ID:</b> AC02
<b>Actors:</b> Human Expert
<b>Preconditions:</b>
The evaluation system has generated the evaluation report, and saved it as "report.json" file.
<b>Flow:</b>
<ol style="list-style-type: none"> <li>1. The Human Expert retrieves the evaluation report from the System.</li> <li>2. The Human Expert opens the report and click on "Extract Results" button.             <ol style="list-style-type: none"> <li>2.1. Human Expert compares the "conflicting labels" to the configured threshold.</li> <li>2.2. Human Expert evaluates the "max conflicting labels" to the threshold.</li> </ol> </li> <li>3. If conflicting labels &gt; threshold or Max consecutive conflicting labels &gt; threshold             <ol style="list-style-type: none"> <li>3.1. The human expert clicks on the "Retrain classifier" button to send a retraining request and asks for a new configuration.</li> </ol> </li> <li>4. else             <ol style="list-style-type: none"> <li>4.1. Classifier passed the evaluation and Human Expert clicks on "Approve Classifier" to denote it as valid.</li> </ol> <p style="margin-left: 20px;">end if</p> </li> <li>5. The Human Expert closes the report.</li> </ol>
<b>Postconditions:</b>
The evaluation report has been evaluated by the system human expert.

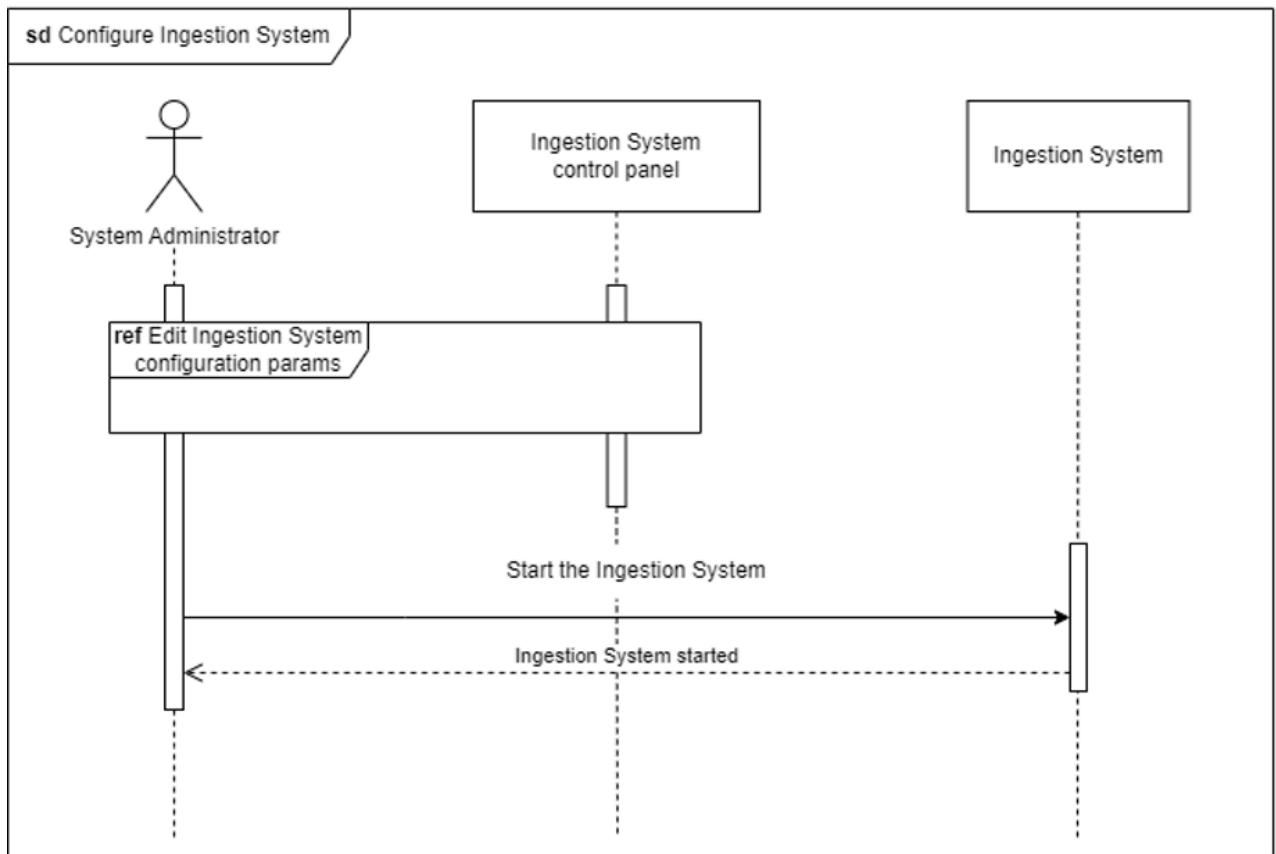
```
evaluationSystemConfiguration.json
{
  "maxNumberOfToleratedErrors" : int,
  "maxNumberOfConsecutiveToleratedErrors" : int,
  "labelsThreshold" : float,
  "testingMode" : bool,
  "evaluationSystemIP" : string,
  "evaluationSystemPort" : int,
  "dbName" : string
}
```

```
evaluationReport.json
{
  "totalErrors" : int,
  "maxConsecutiveErrors" : int
}
```

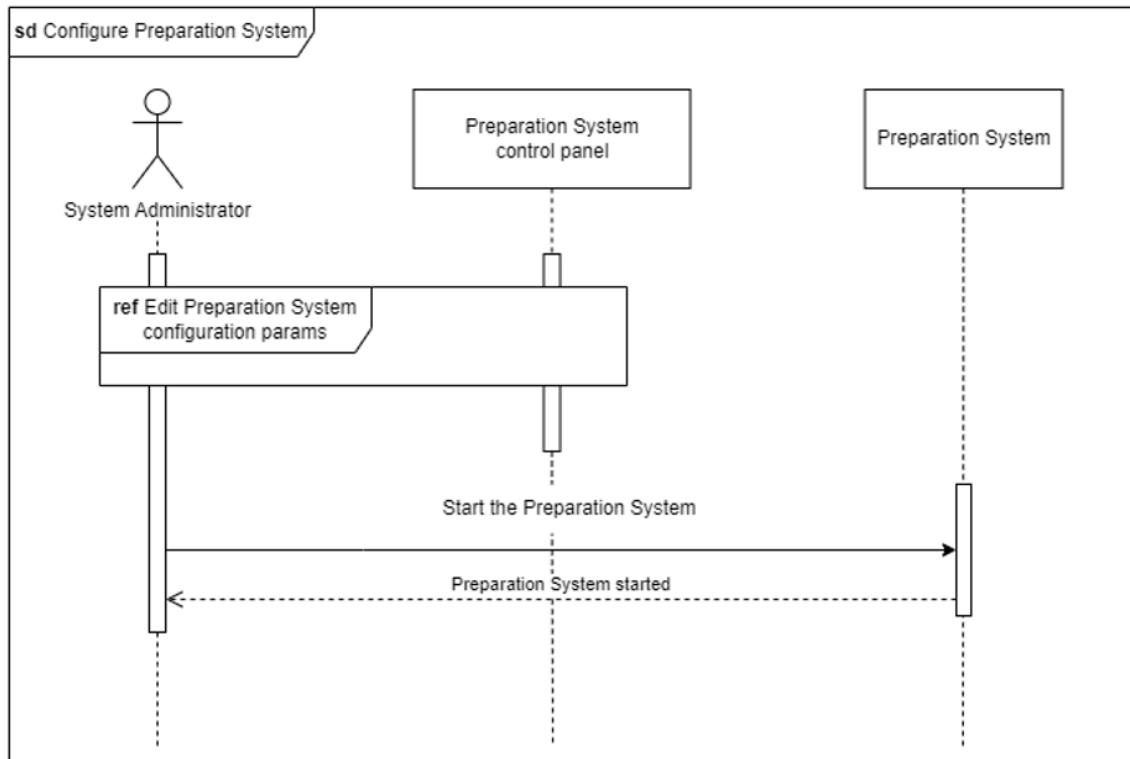
```
sessionLabel.json
{
  "uuid" : string,
  "label" : string
}
```

# Sequence Diagrams

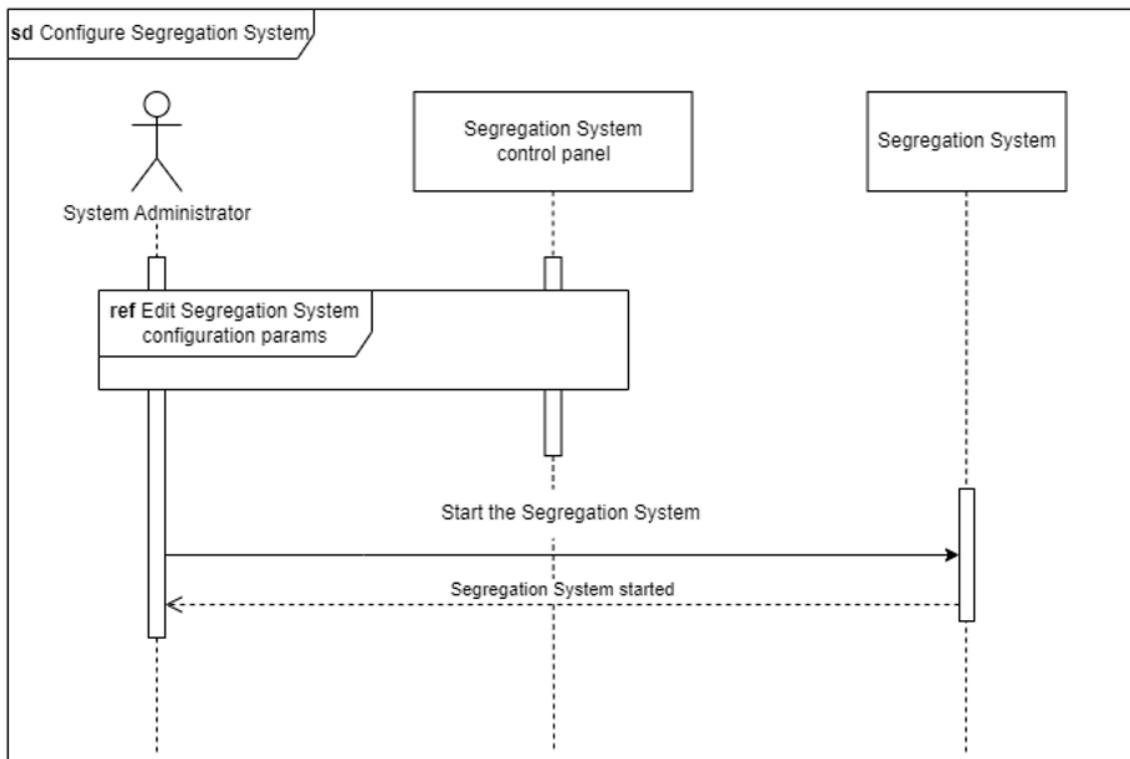
## Ingestion System (Daka)

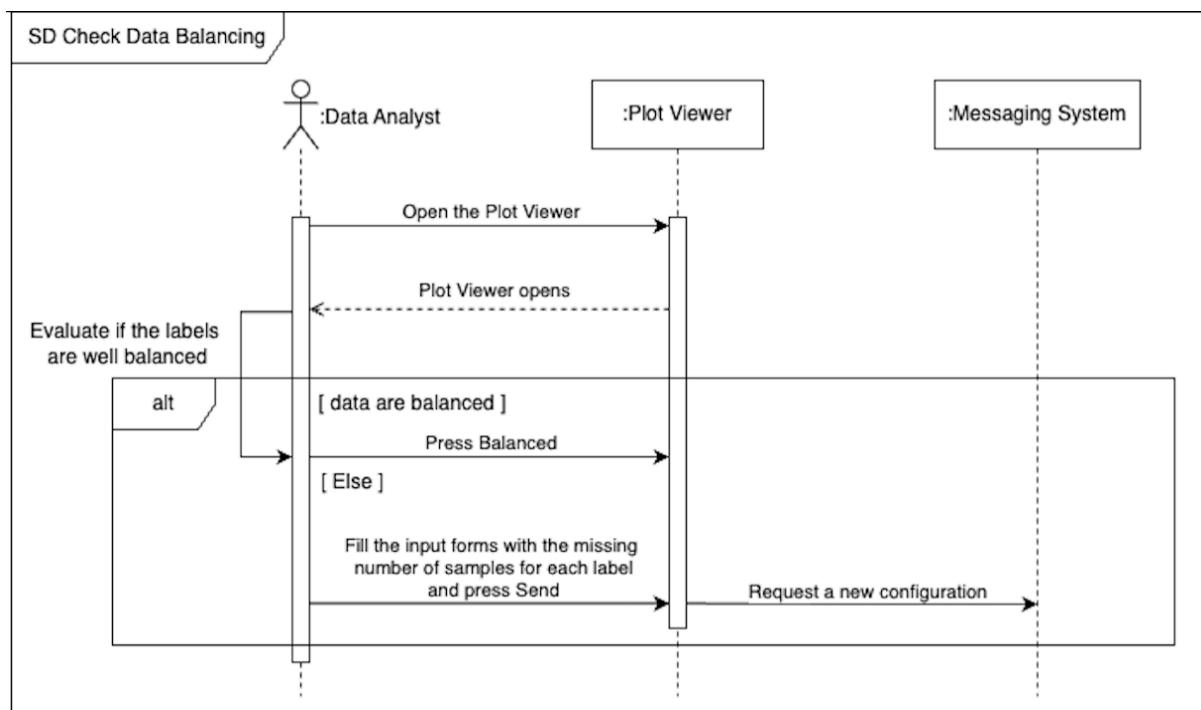
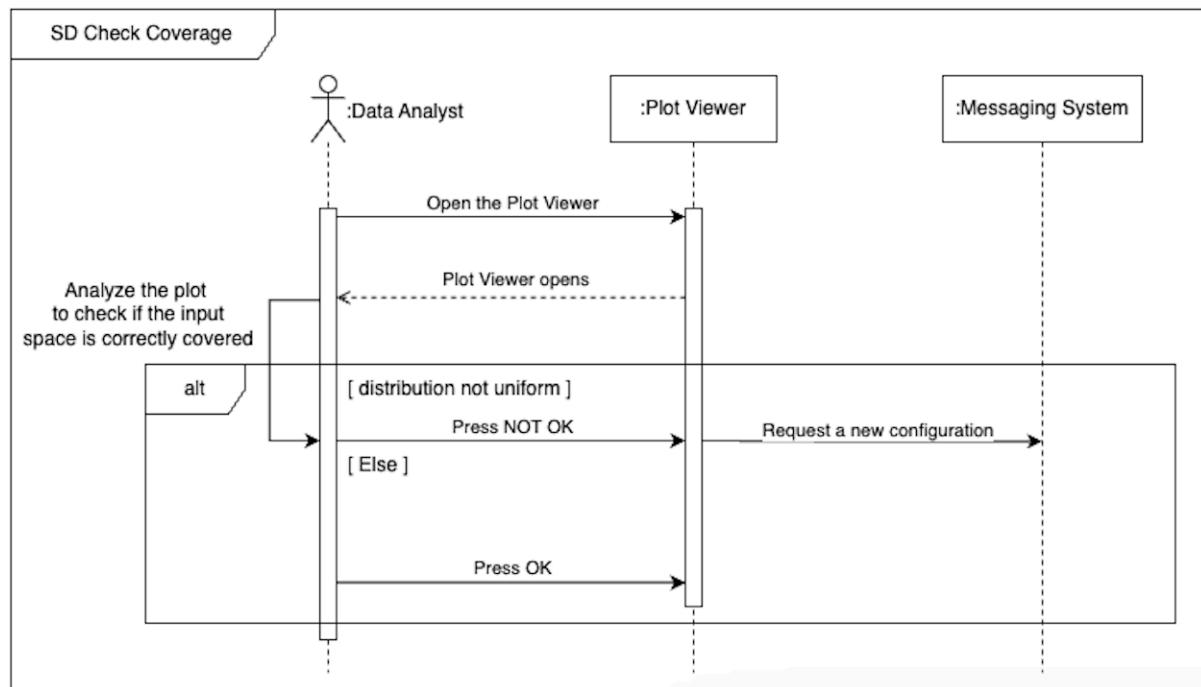


## Preparation System (Daka)

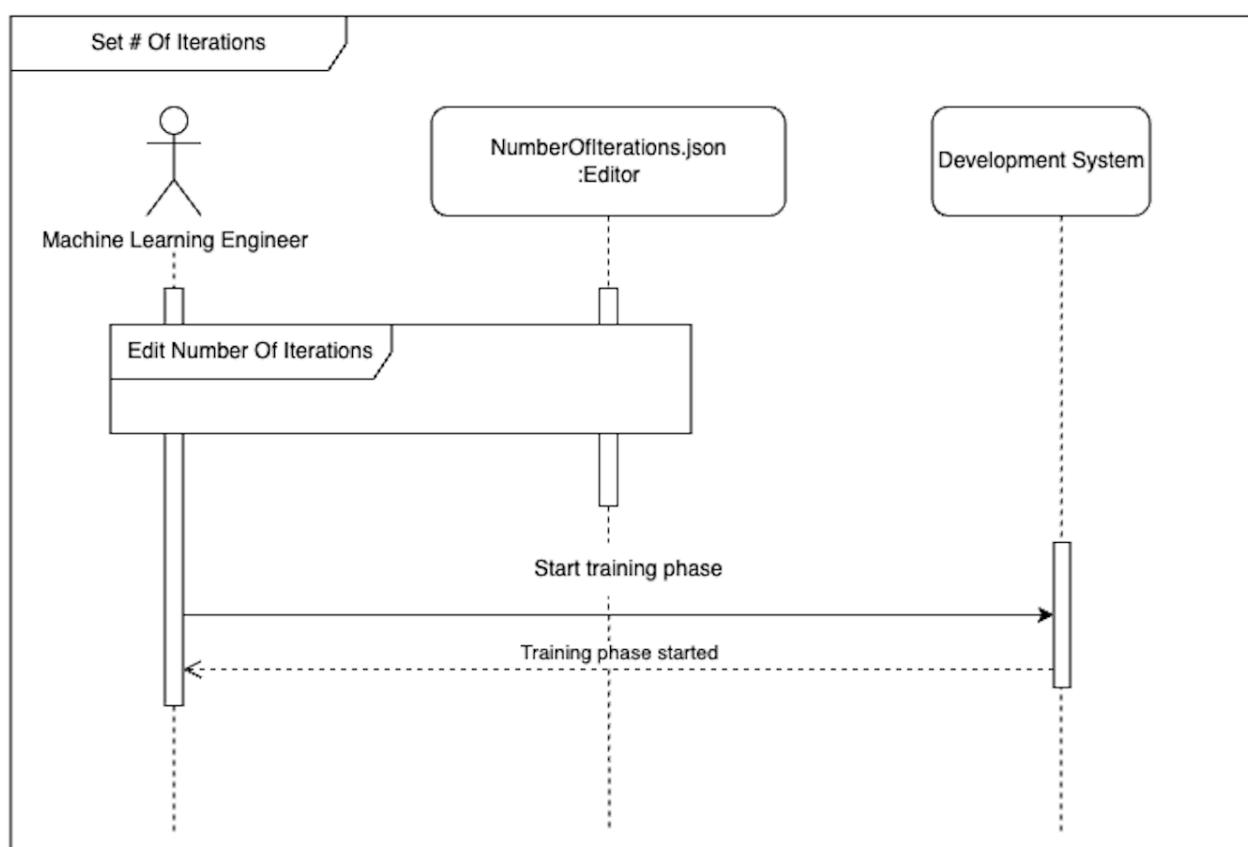
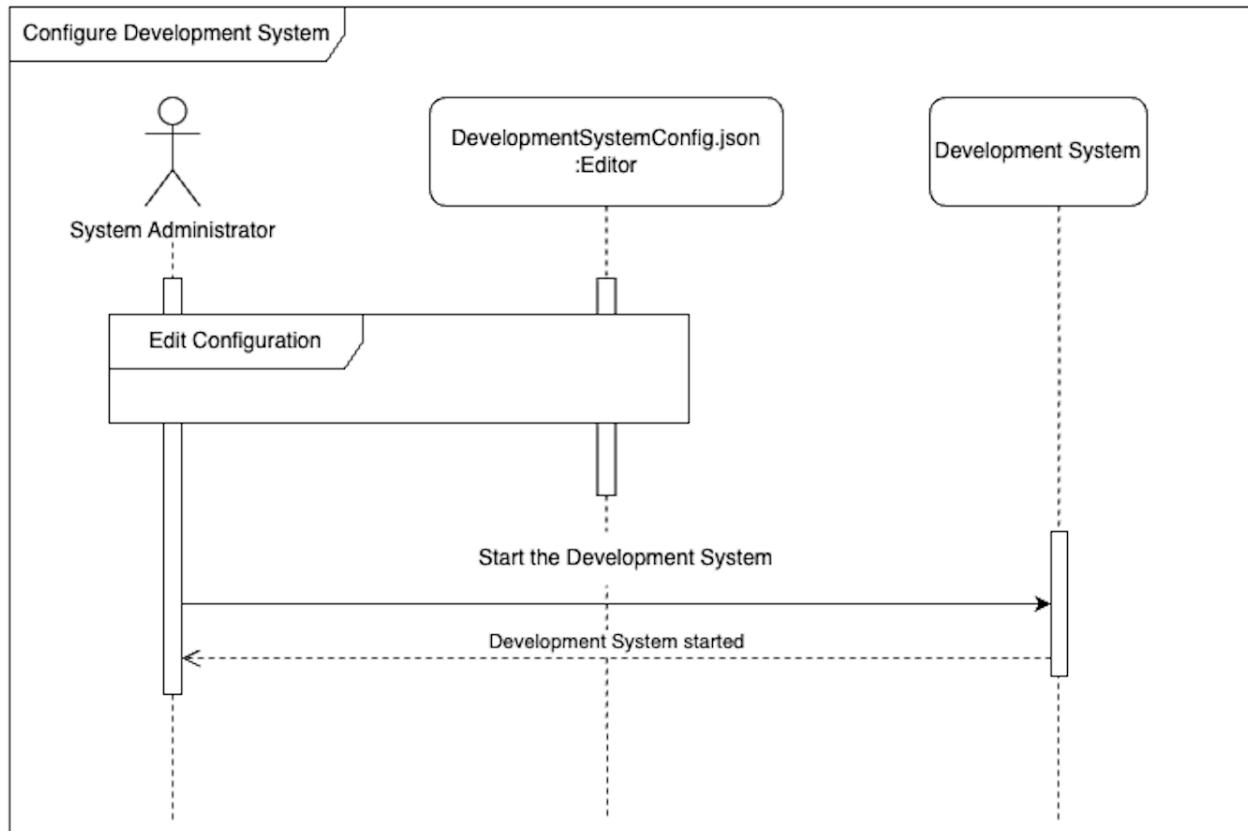


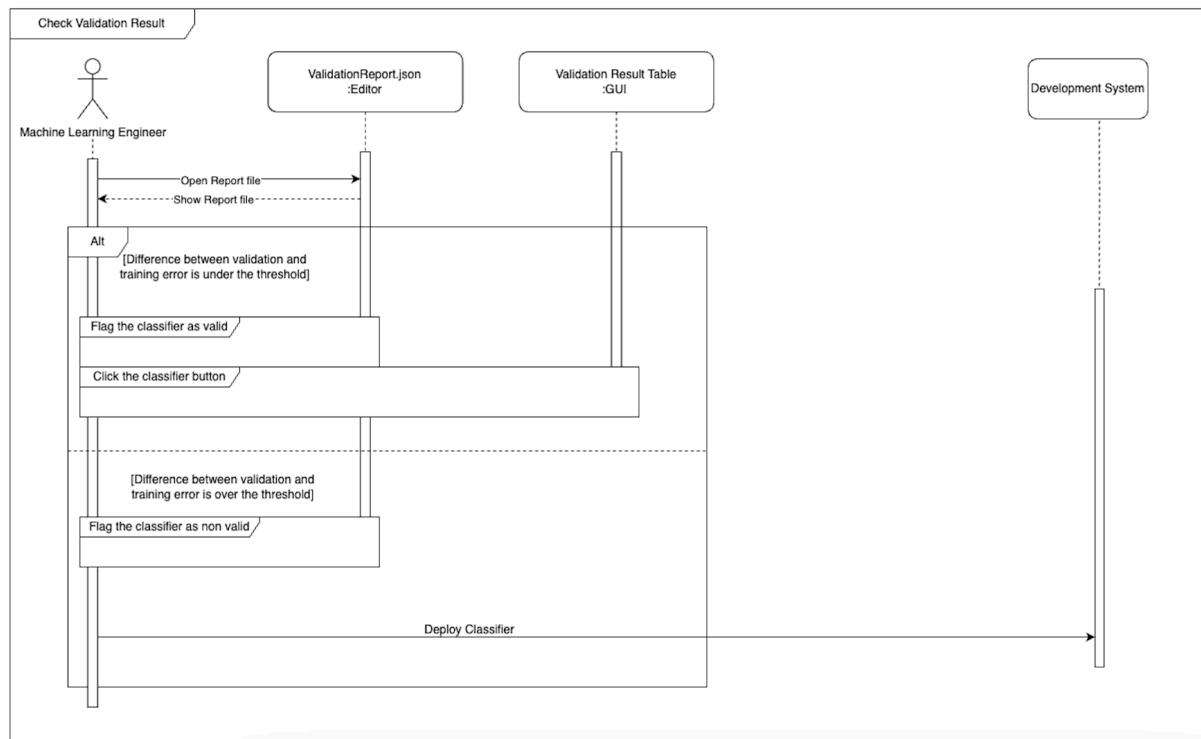
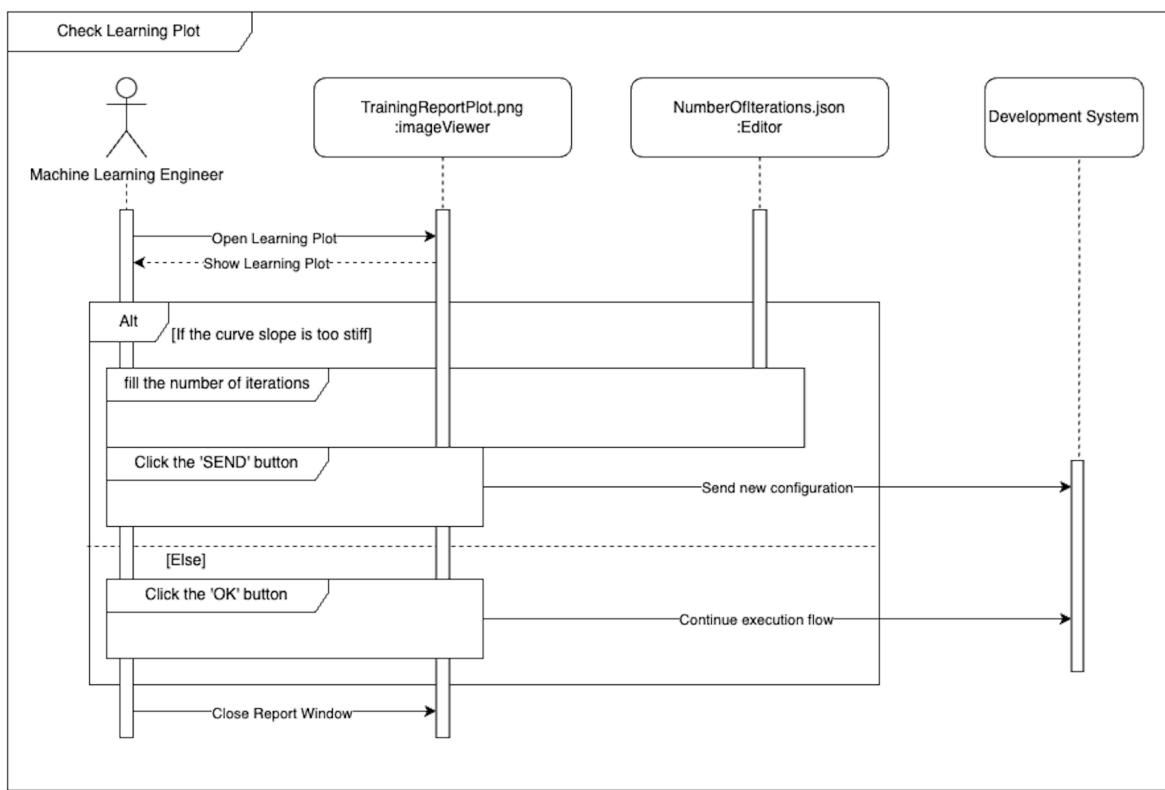
## Segregation System (Pantè, Daka)



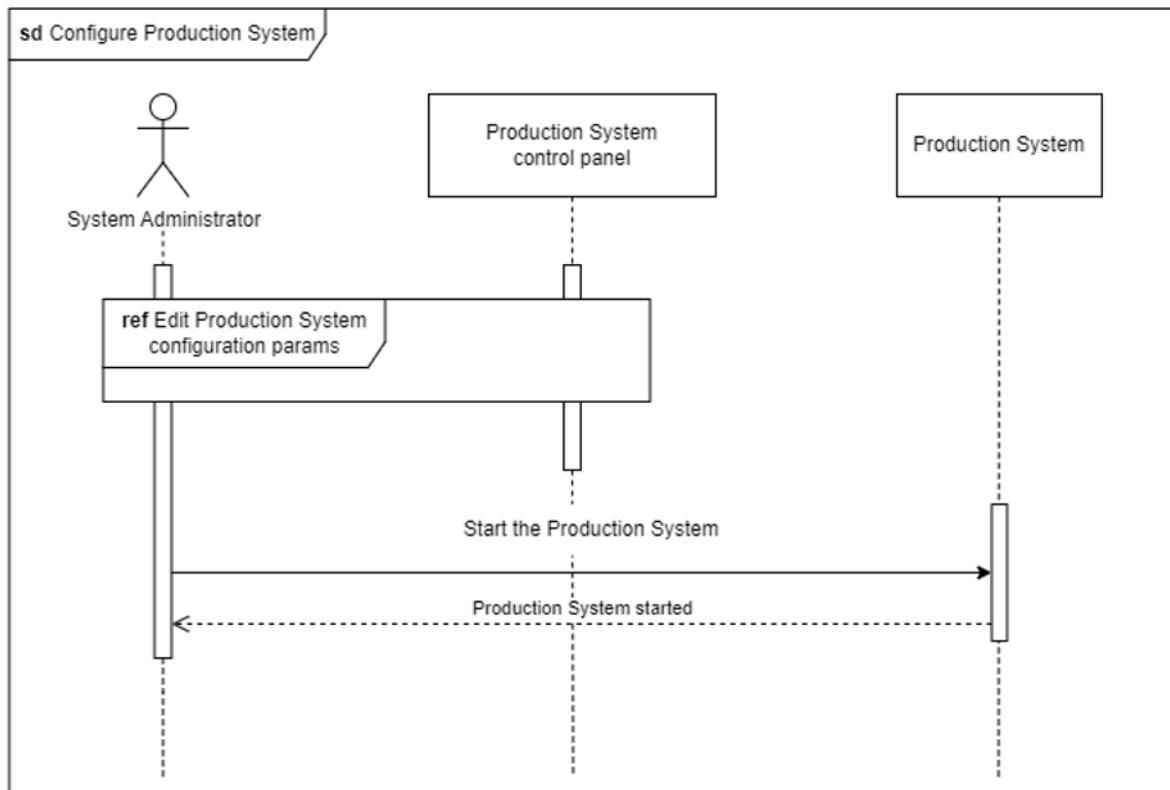


## Development System (Bataloni)

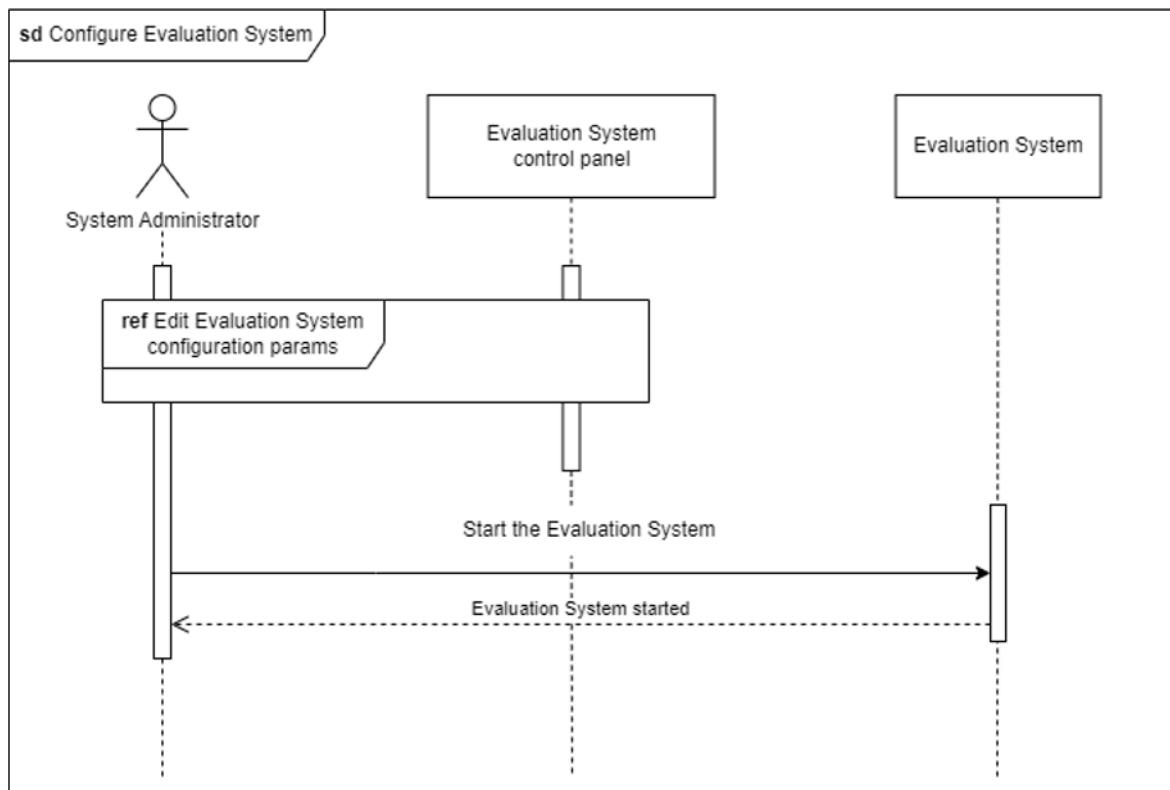


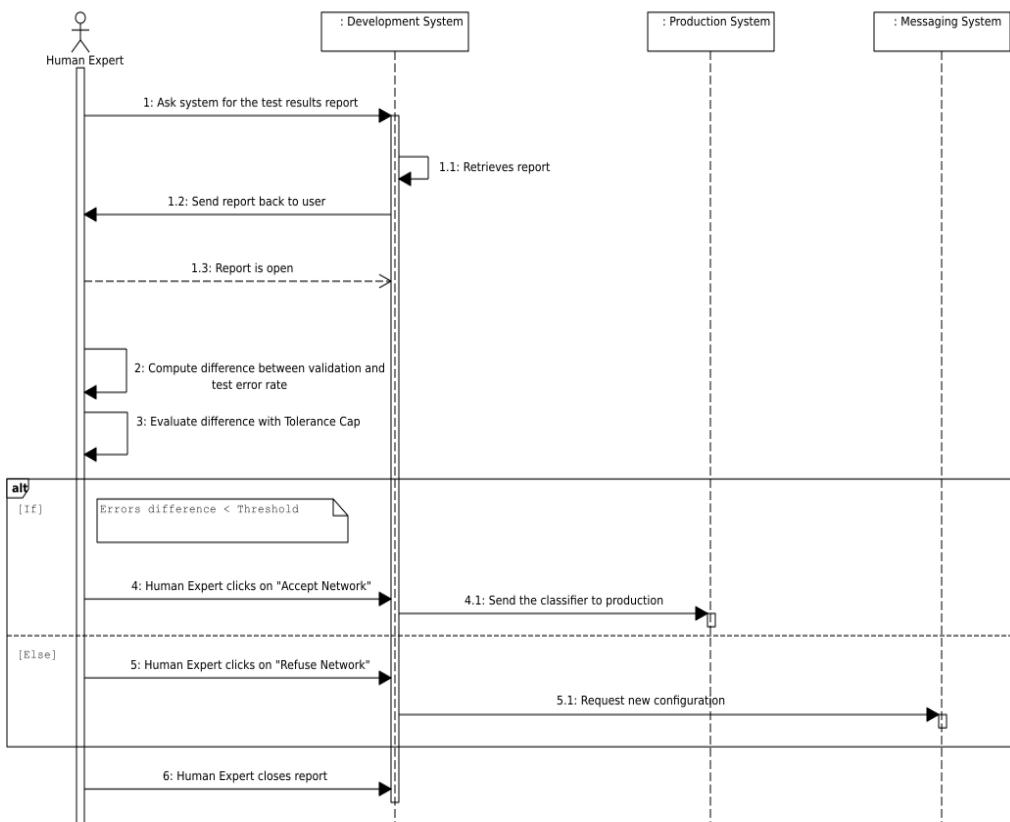
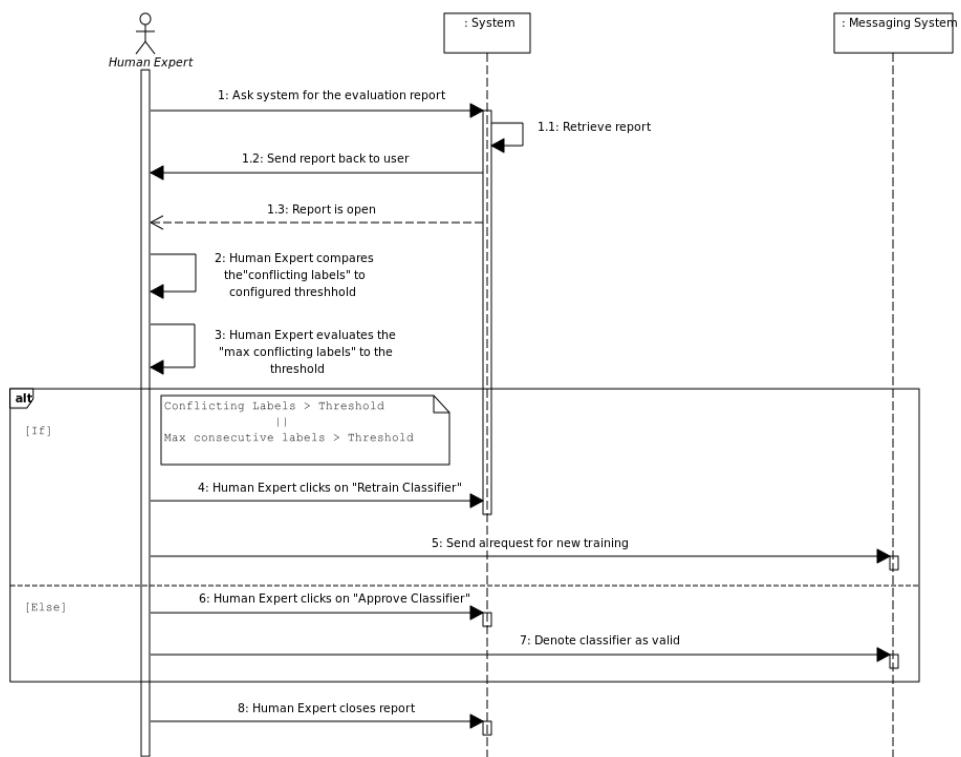


## Production System (Daka)



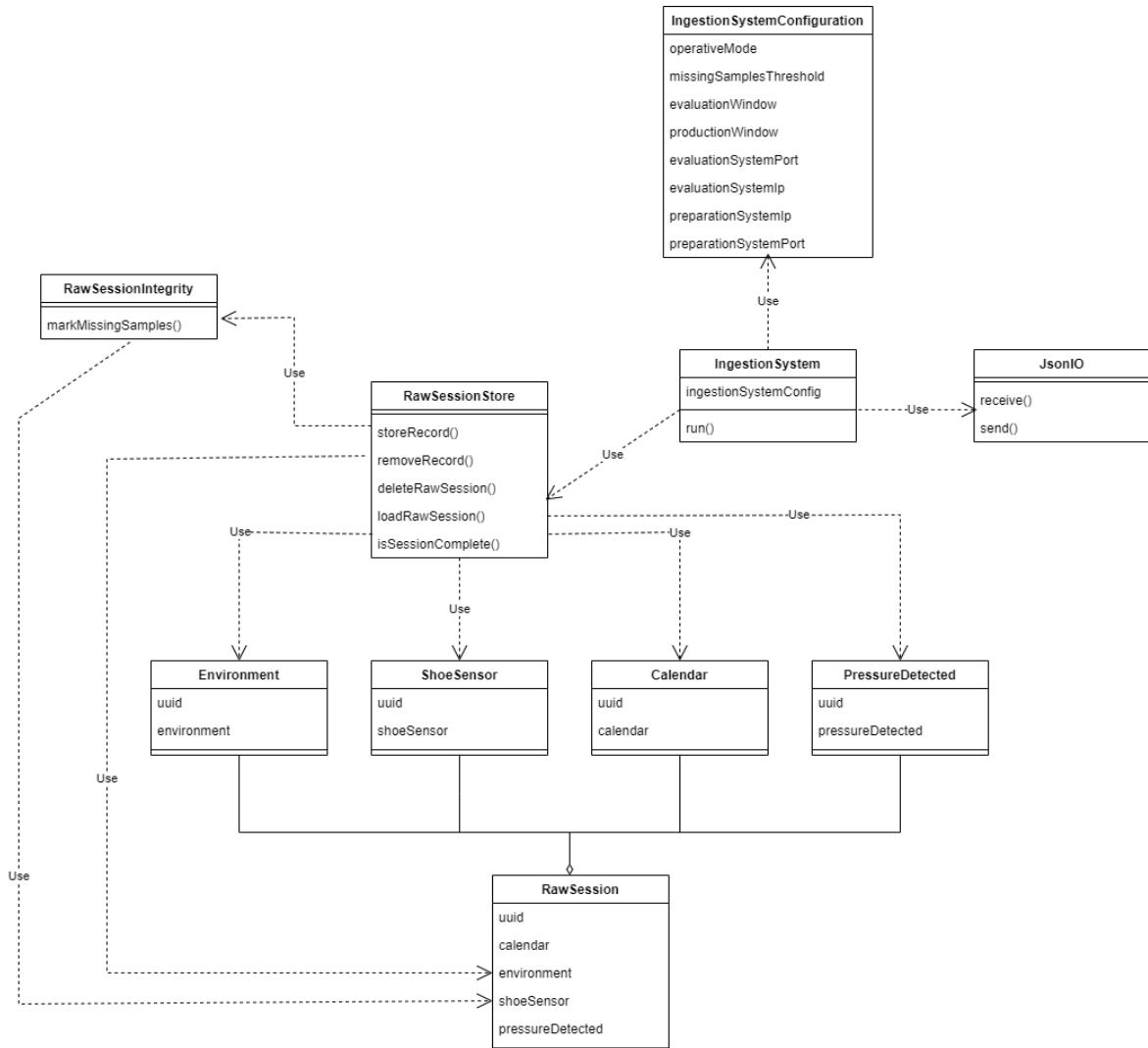
## Evaluation System (Bidani, Daka)



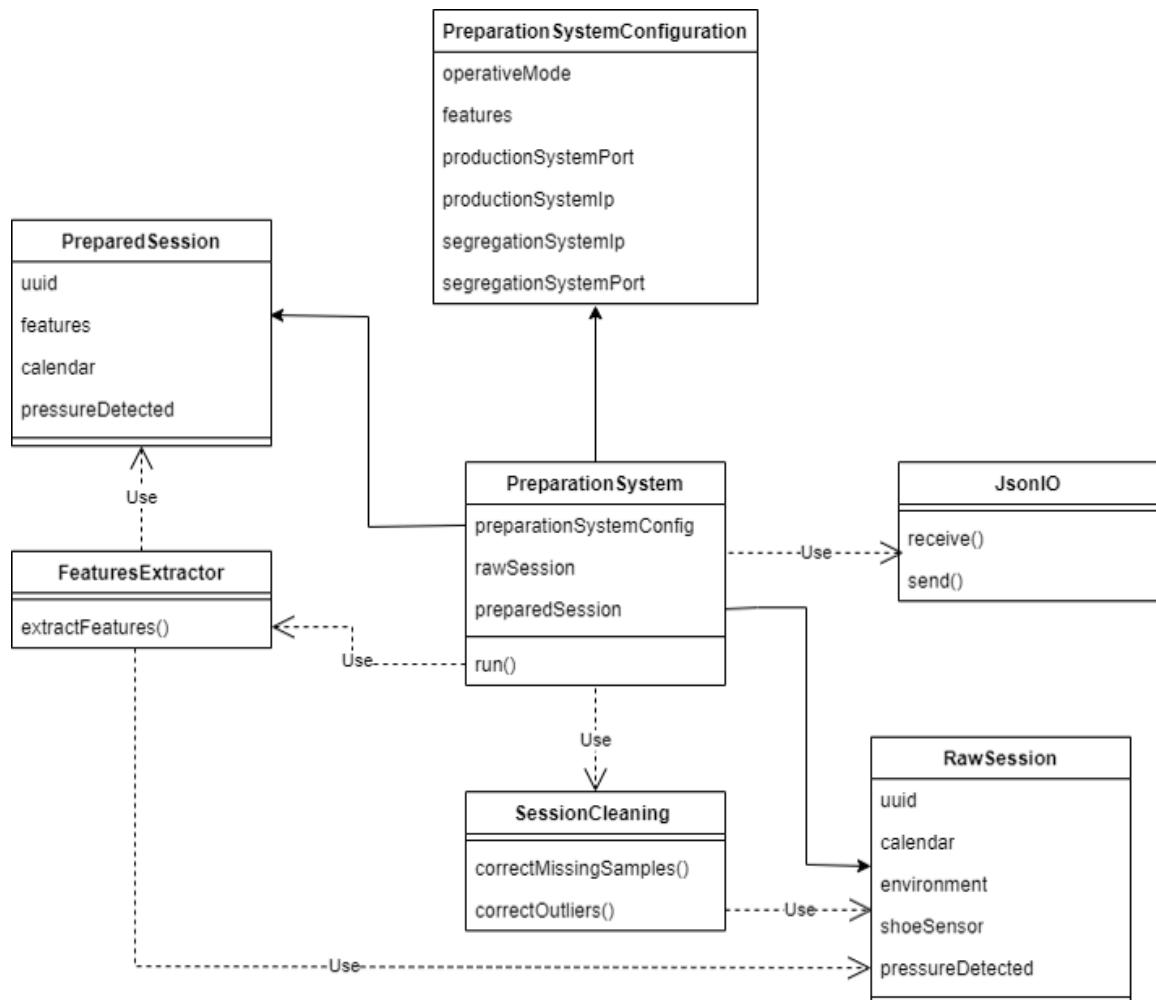


# Class Diagrams (Analysis)

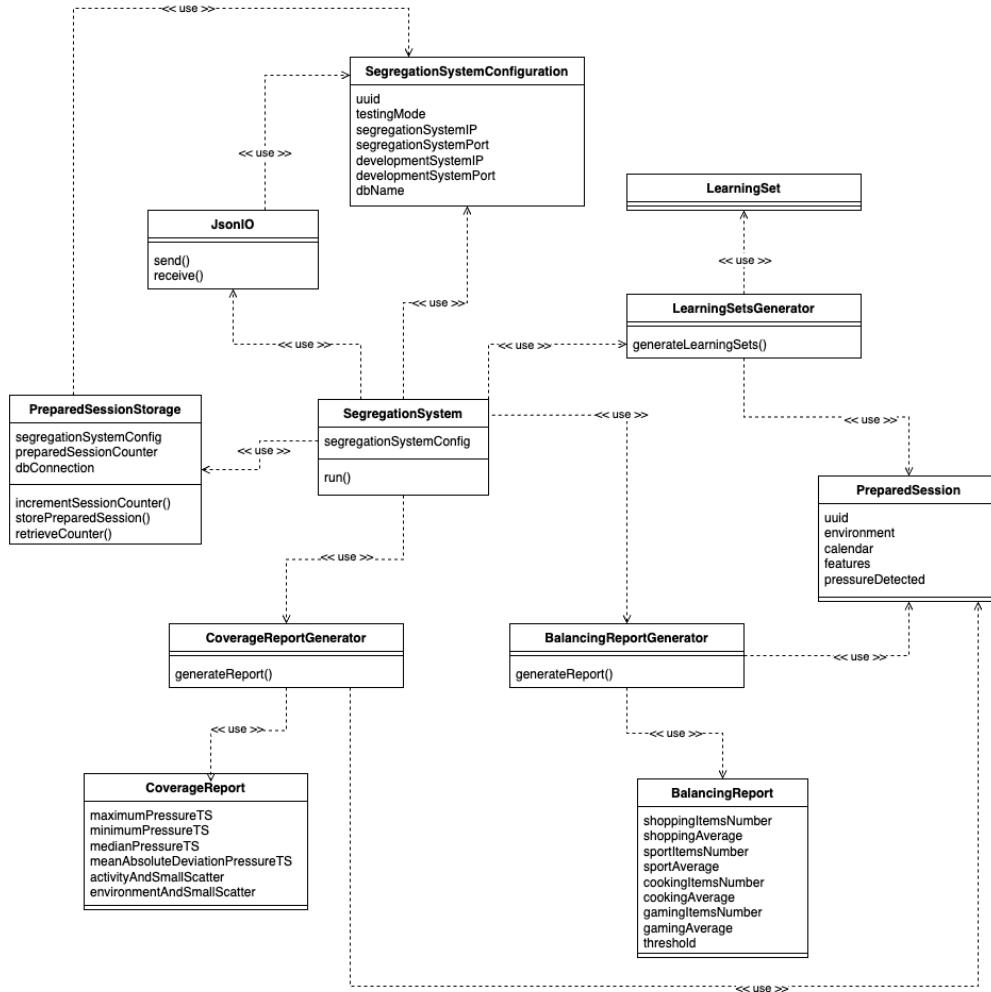
## Ingestion System (Daka)



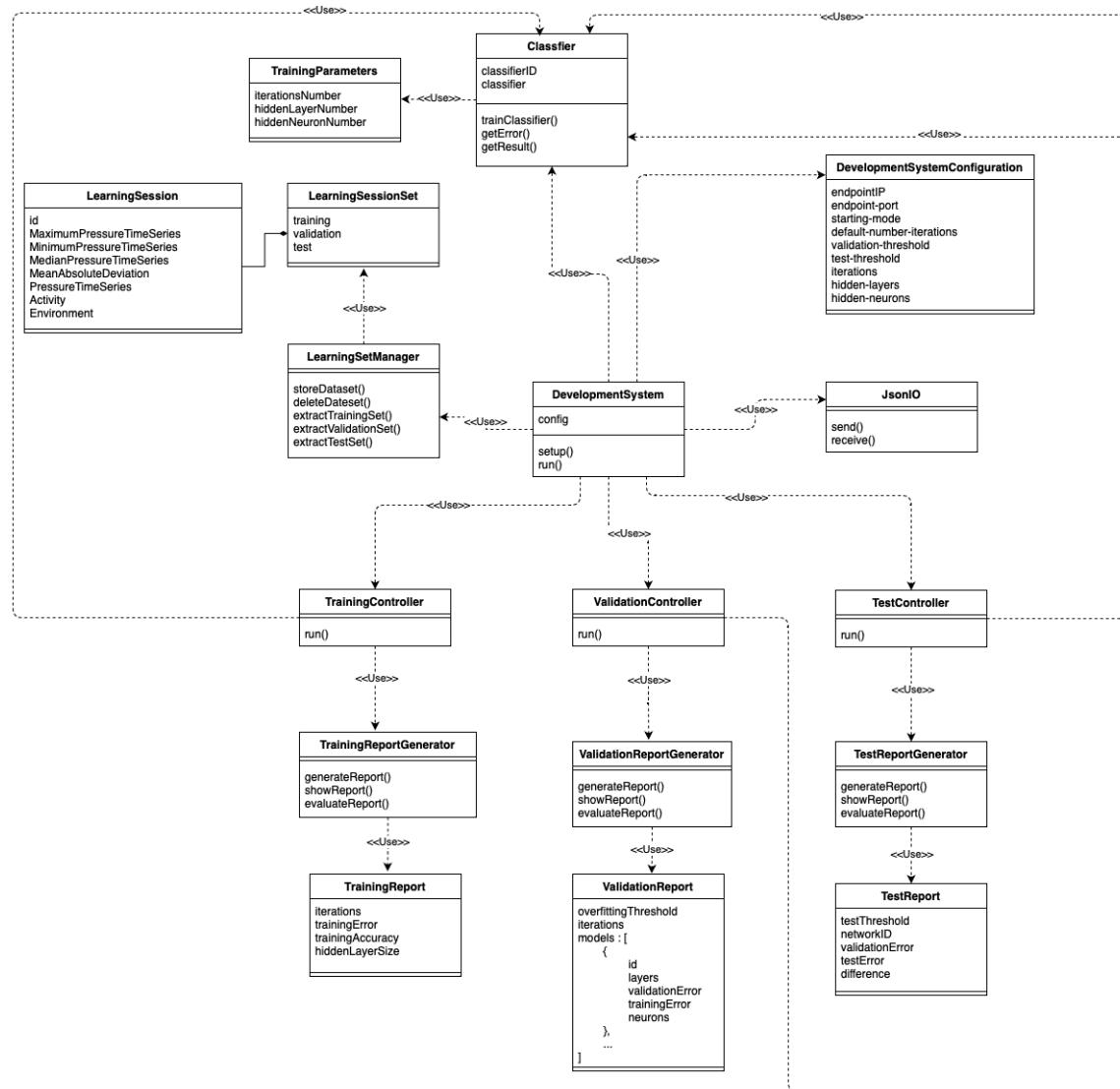
## Preparation System (Daka)



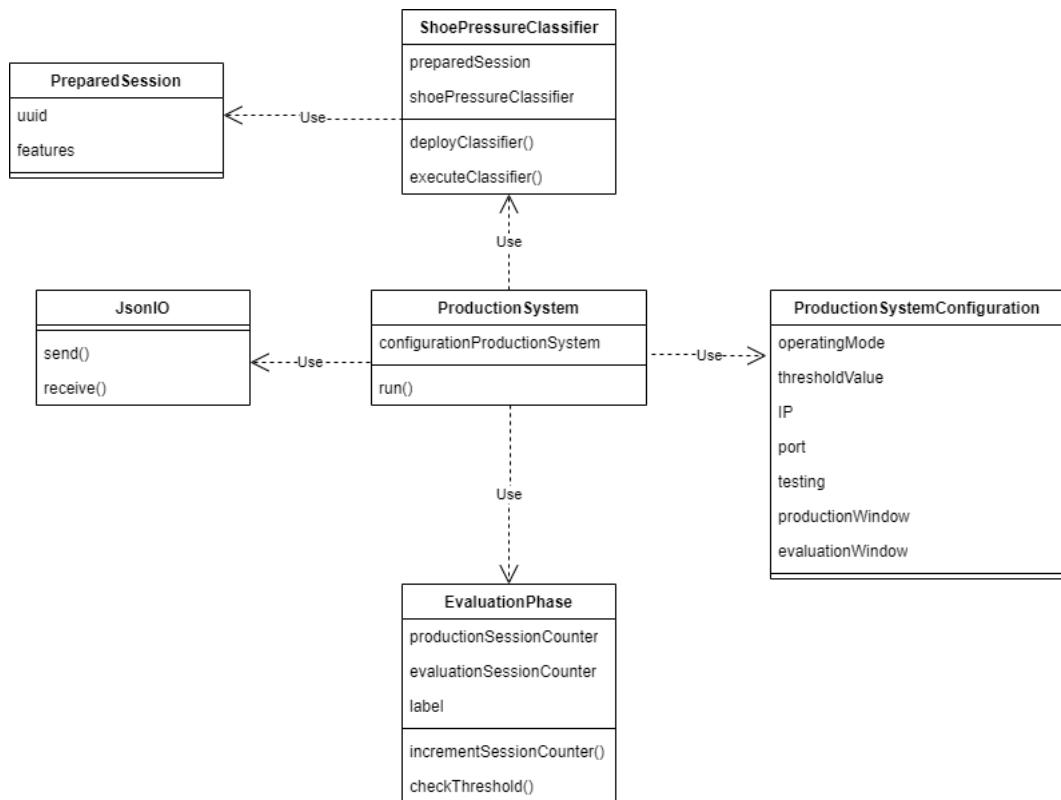
# Segregation System (Pantè)



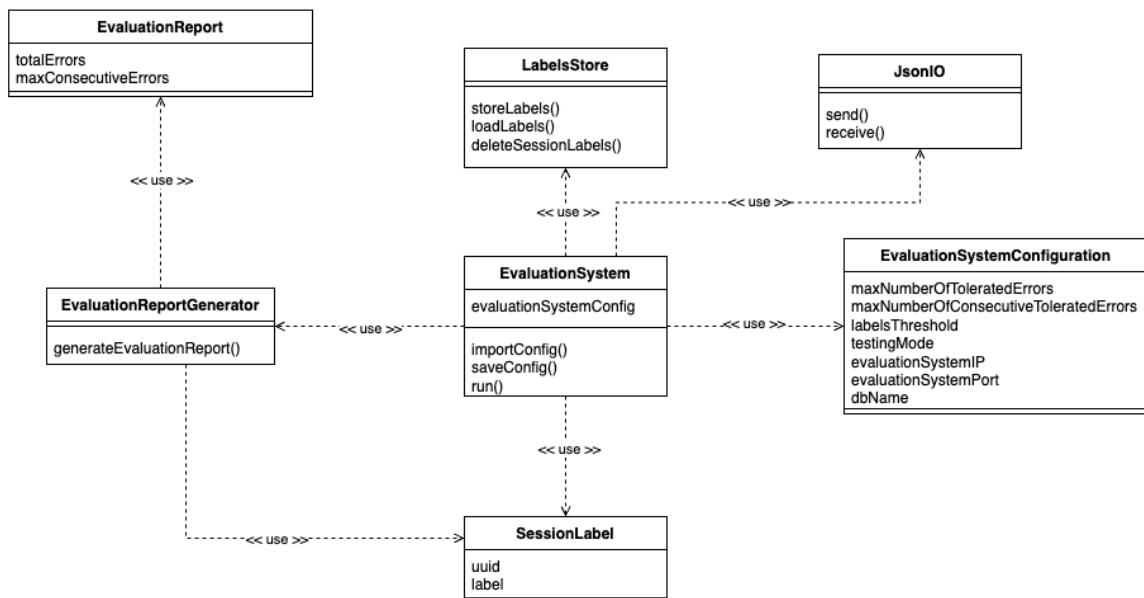
# Development System (Bataloni)



## Production System (Daka)



## Evaluation System (Pantè)



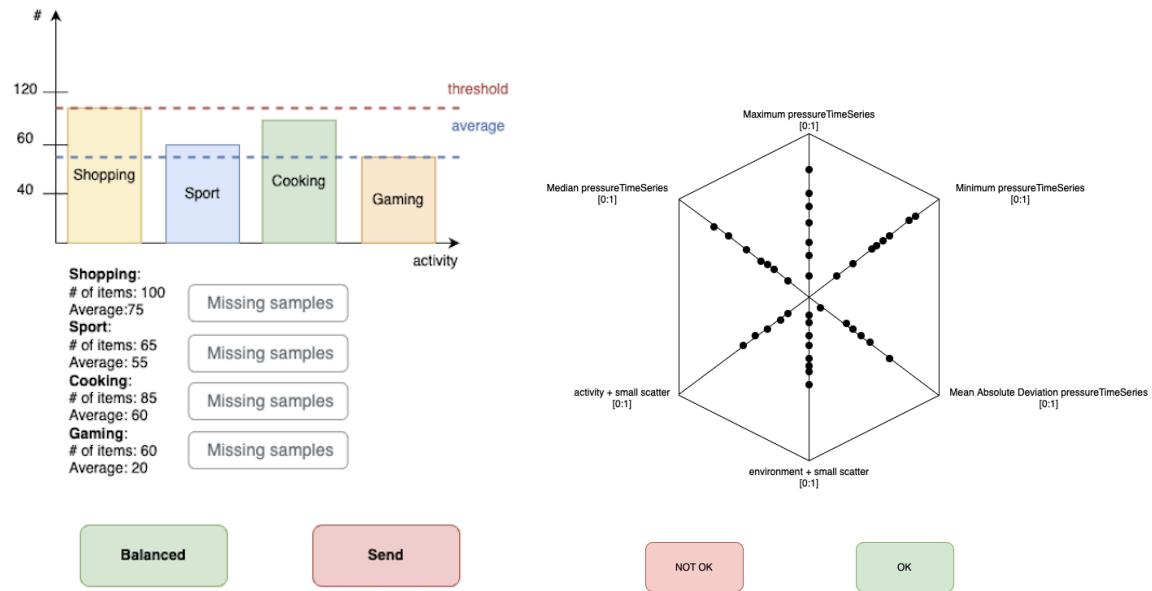
# Mockups

## Configuration System (Pantè)

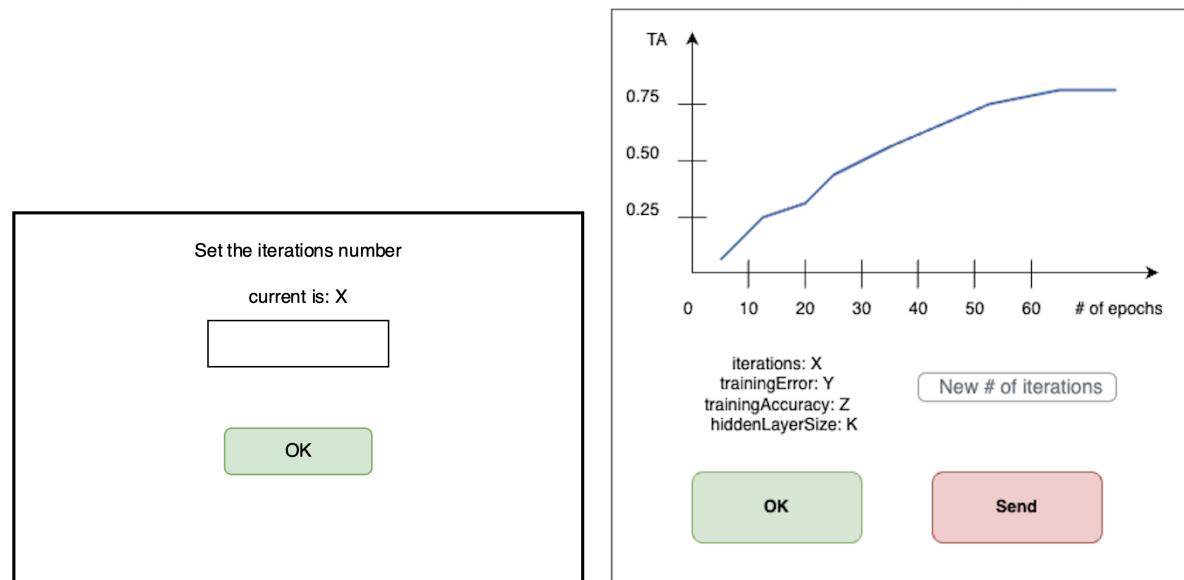
<b>Ingestion System</b>	<b>Preparation System</b>	<b>Segregation System</b>	<b>Development System</b>												
<b>Operative Mode</b> production	<b>Operative Mode</b> production	<b>Stage</b> store	<b>Starting Mode</b> waiting												
<b>Preparation System IP</b> 127.0.0.1	<b>Segregation System IP</b> 10.8.0.3	<b>Segregation System IP</b> 0.0.0.0	<b>Ongoing Validation</b> false												
<b>Preparation System Port</b> 5000	<b>Segregation System Port</b> 6000	<b>Segregation System Port</b> 6000	<b>Stage</b> waiting												
<b>Evaluation System IP</b> 10.8.0.7	<b>Production System IP</b> 10.8.0.2	<b>Development System IP</b> 10.8.0.2	<b>src IP</b> 0.0.0.0												
<b>Evaluation System Port</b> 6000	<b>Production System Port</b> 6001	<b>Development System Port</b> 6000	<b>src Port</b> 6000												
<b>Production Window</b> 5	<b>Max Value</b> 3	<b>Preparation System IP</b> 10.8.0.6	<b>dest IP</b> 127.0.0.1												
<b>Evaluation Window</b> 5	<b>Min value</b> 0	<b>Preparation System Port</b> 5000	<b>dest Port</b> 6001												
<b>Missing Sample Threshold</b> 1	<b>Features:</b>	<b>Database Name</b> segregation.db	<b>Iterations Number</b> 5												
<b>Database Name</b> mobility.db	<table border="1"><tr><td><b>Environment</b></td><td><b>Calendar</b></td></tr><tr><td>1 Slippery</td><td>1 Shopping</td></tr><tr><td>2 Plain</td><td>2 Sport</td></tr><tr><td>3 Slope</td><td>3 Cooking</td></tr><tr><td>4 House</td><td>4 Gaming</td></tr><tr><td>5 Track</td><td></td></tr></table>	<b>Environment</b>	<b>Calendar</b>	1 Slippery	1 Shopping	2 Plain	2 Sport	3 Slope	3 Cooking	4 House	4 Gaming	5 Track		<b>Max Sessions</b> 10	<b>Hidden Layer Size Range</b> <div style="display: flex; align-items: center;"><input type="range" value="0"/><span>0</span><span>10</span></div>
<b>Environment</b>	<b>Calendar</b>														
1 Slippery	1 Shopping														
2 Plain	2 Sport														
3 Slope	3 Cooking														
4 House	4 Gaming														
5 Track															
		<b>Train Set Size</b> 0.7	<b>Hidden Neuron Per Layer Range</b> <div style="display: flex; align-items: center;"><input type="range" value="0"/><span>0</span><span>200</span></div>												
		<b>Validation Set Size</b> 0.2	<b>Variation Step</b> 1												
		<b>Test Set Size</b> 0.1	<b>Overfitting Threshold</b> 0.25												
			<b>Validation Tolerance</b> 0.3												
<b>Save</b>	<b>Save</b>	<b>Save</b>	<b>Save</b>												
<b>Discard</b>	<b>Discard</b>	<b>Discard</b>	<b>Discard</b>												

<b>Production System</b>	<b>Evaluation System</b>
<b>Evaluation Phase</b> true	<b>Stage</b>
<b>Classifier Deployed</b> false	<b>Evaluation System IP</b> 0.0.0.0
<b>src IP</b> 0.0.0.0	<b>Evaluation System Port</b> 6000
<b>src Port</b> 6001	<b>Ingestion System IP</b>
<b>Evaluation System IP</b> 10.8.0.7	<b>Ingestion System Port</b>
<b>Evaluation System Port</b> 6000	<b>Production System IP</b>
<b>Client System IP</b> 172.123.0.0	<b>Production System Port</b>
<b>Client System Port</b> 6000	<b>Database Name</b> evaluation.db
<b>Messaging System IP</b> 172.23.0.0	<b>Threshold</b> 20
<b>Messaging System Port</b> 6000	<b>Max Number Of Tolerared Errors</b> 10
	<b>Max Number Of Tolerared Consecutive Errors</b> 4
<b>Save</b>	<b>Save</b>
<b>Discard</b>	<b>Discard</b>

## Segregation System (Pantè, Bataloni)



## Development System (Bataloni, Pantè, Daka, Bidani)



RESULTS				
ID	Validation Error	Training Error	# of layers	# of neurons
#NN002	0,3	0,1	7	30
#NN001	0,6	0,1	6	50
#NN004	0,7	0,3	8	25
#NN005	1,1	0,5	5	40
#NN003	1,2	0,6	4	60

Overfitting tolerance:  
0,5

None

Select the network:

NN001 NN002 NN003 NN004 NN005

Tolerance Cap: 25%

Winner Network Validation Error Test Error Difference

#NN03 0,3 0,25 0,05

Accept Network

Refuse Network

## Evaluation System (Bidani)

Classifier Evaluation Error Report		
Expert Label Class	Classifier Output Class	Result
Anomalous	Regular	✗
Regular	Regular	✓
Regular	Anomalous	✗
Anomalous	Regular	✗
Anomalous	Anomalous	✓
Regular	Regular	✓
Regular	Anomalous	✗

**Max Numbers of Errors tolerated (TH1) :**

3

**Max Numbers of Consecutive Errors tolerated (TH2) :**

2

**Total errors : 4**

**Max consecutive errors : 2**

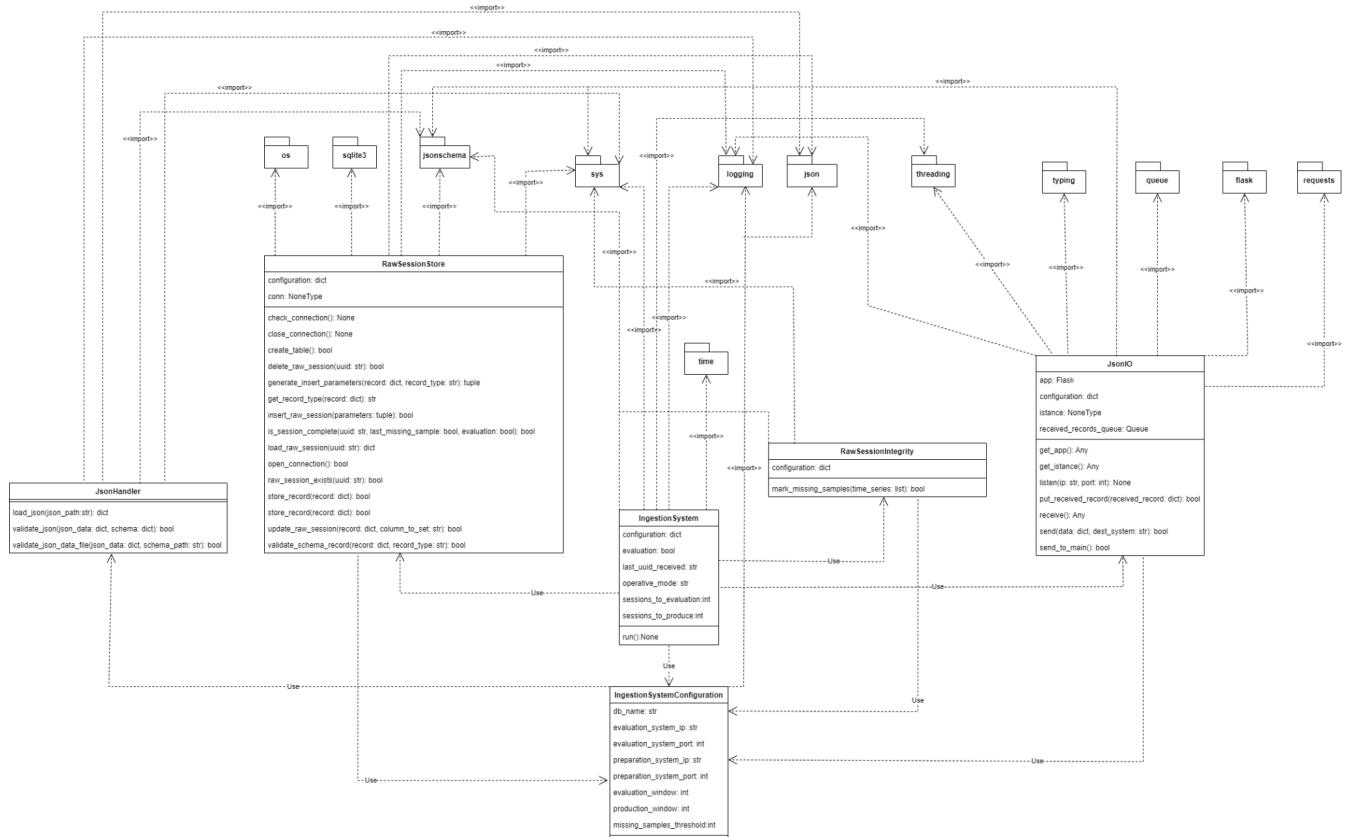
**TH1 exceeded (4>3)**

**TH2 satisfied (2=2)**

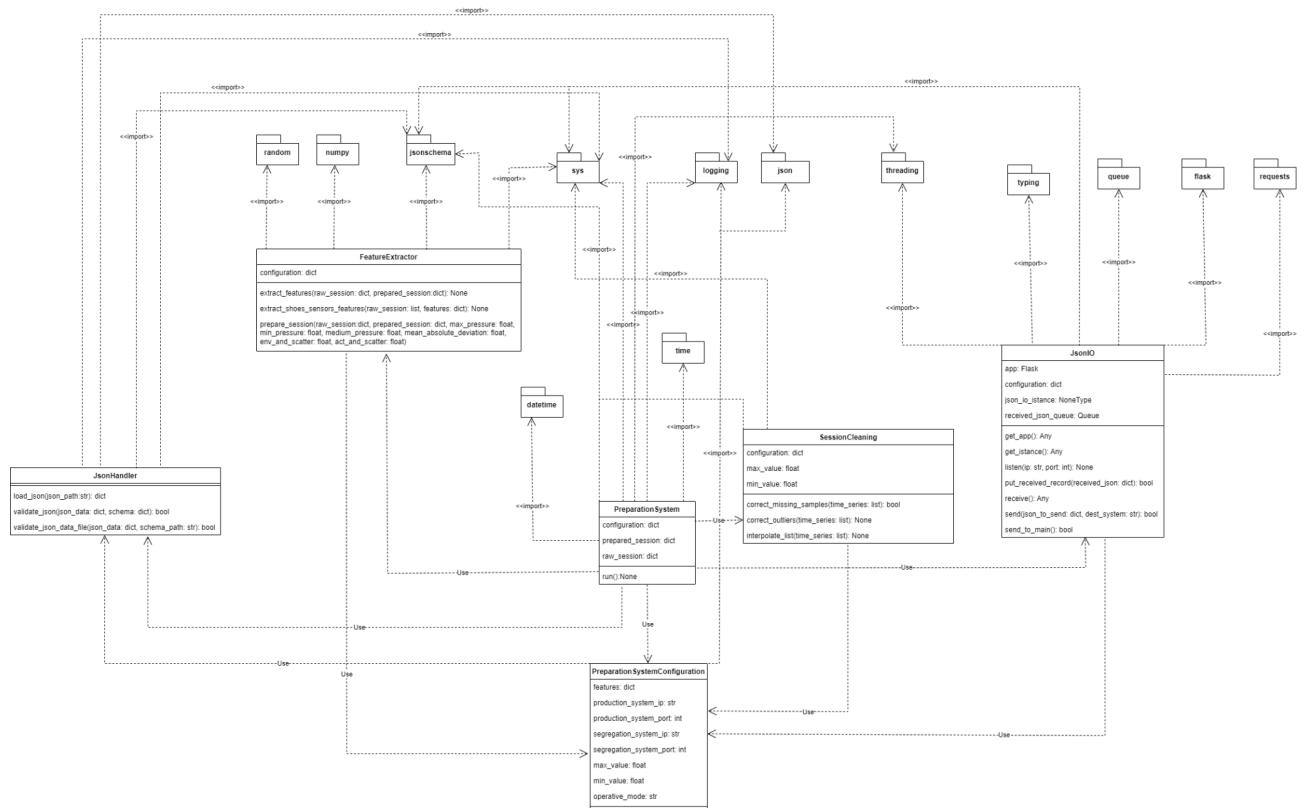
Extract results
Retrain Classifier
Approve classifier

# Class Diagrams (Design)

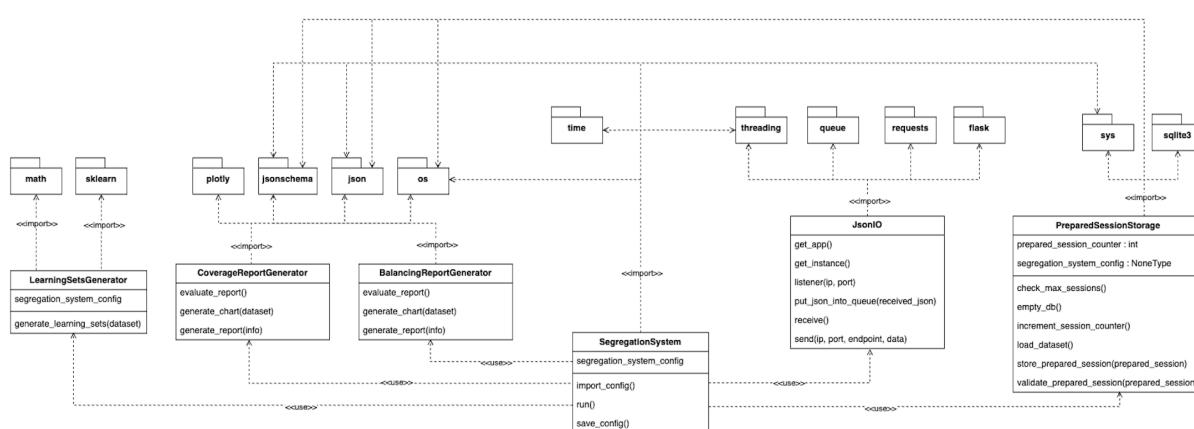
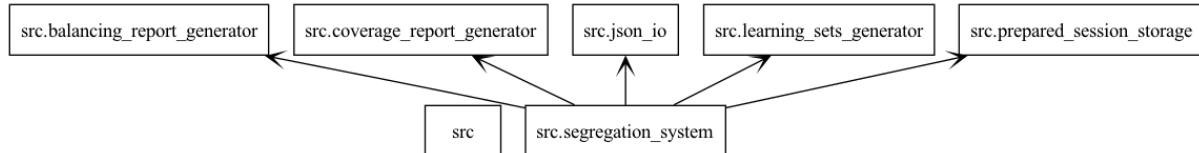
## Ingestion System (Daka)



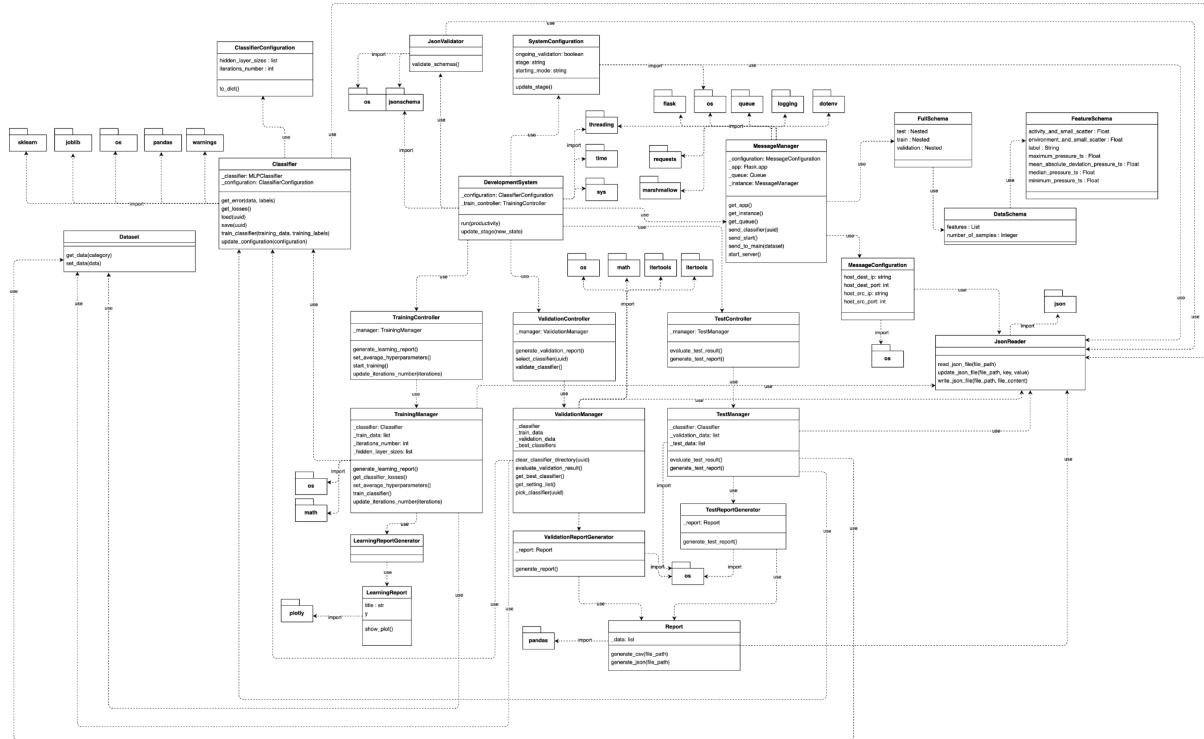
## Preparation System (Daka)



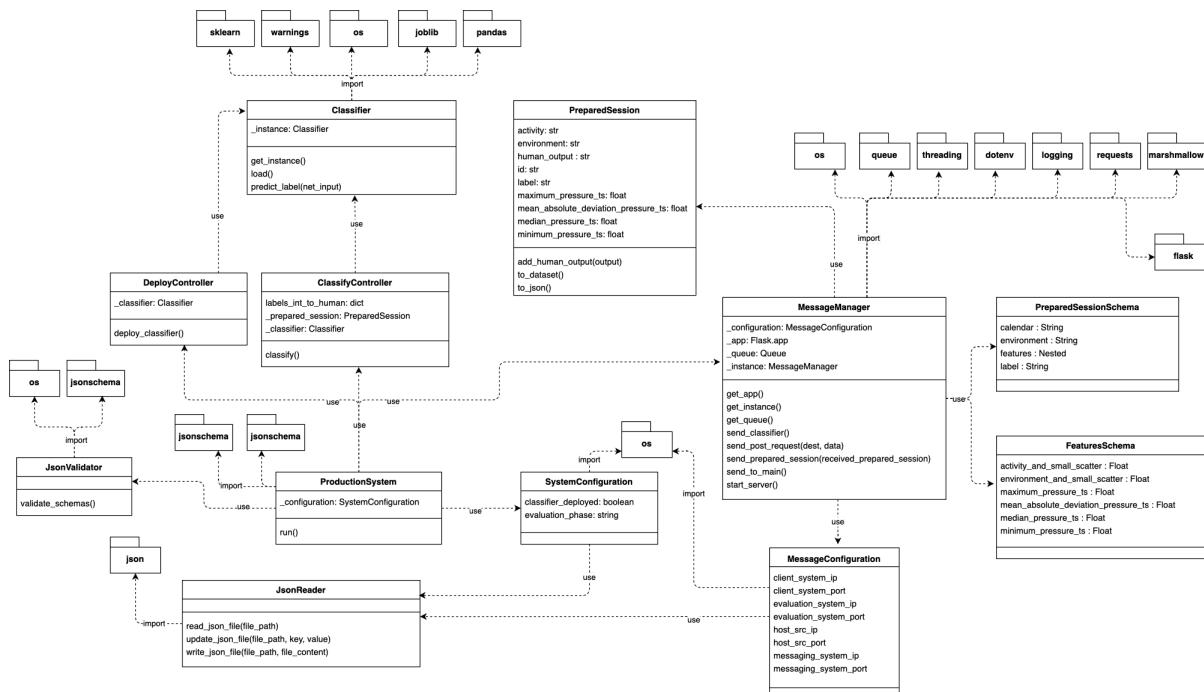
## Segregation System (Pantè)



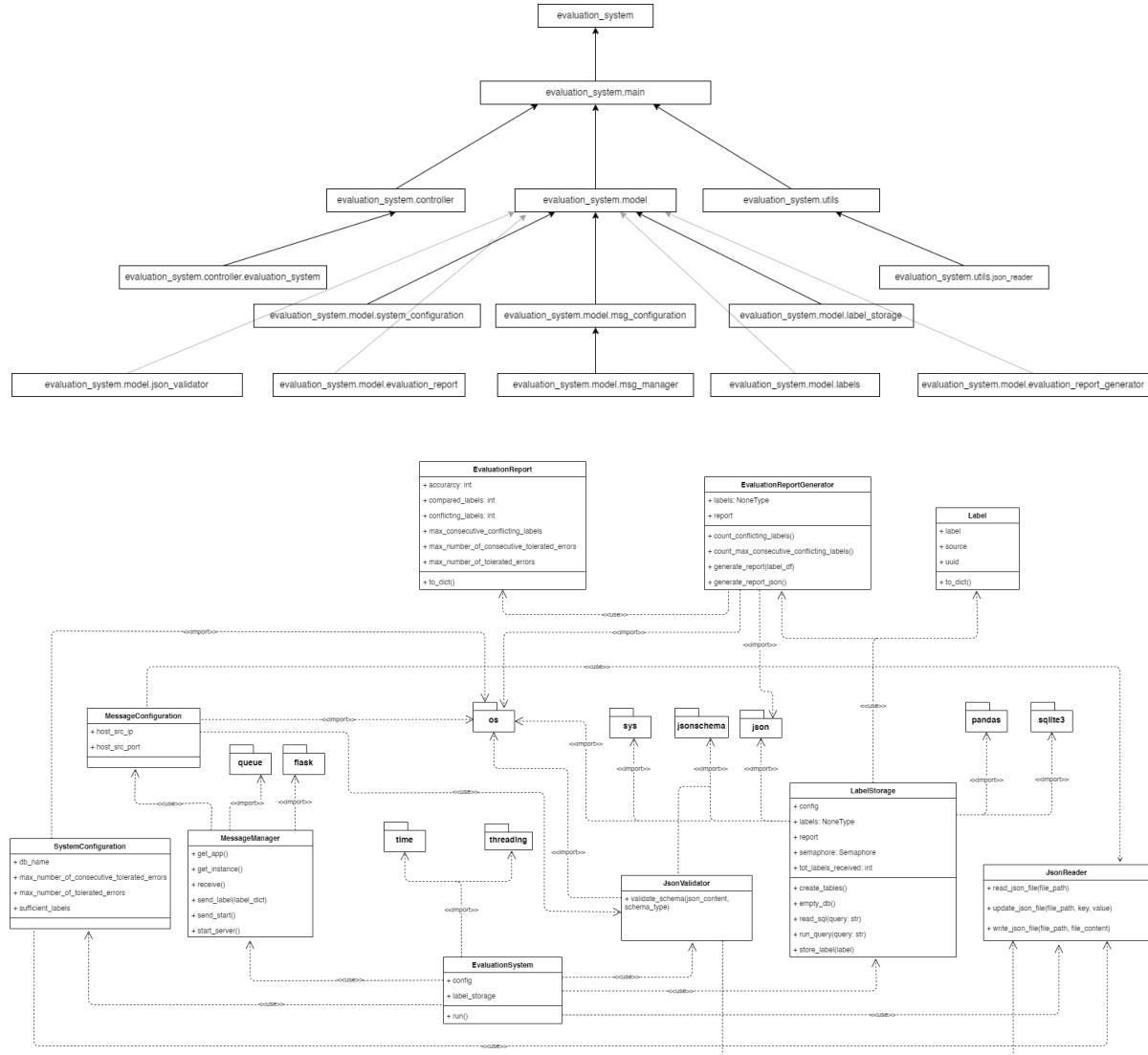
## Development System (Bataloni)



## Production System (Bataloni)



# Evaluation System (Bidani)



# Testing (Non-Resiliency)

## Ingestion System (Daka)

ID	Input Error	Consequence	Score
I1	<ul style="list-style-type: none"> <li>- Missing records (calendar, environment)</li> <li>- Missing label in development or evaluation phase</li> </ul>	The raw session is not completed so it is discarded	4
I2	Missing pressure time series samples (under threshold)	The session is marked and sent to the Preparation System (solved by the Preparation System)	1
I3	Missing pressure time series samples(over threshold)	The session is discarded	4
I4	Duplicated records (same uuid and same values)	The record is overwritten with the same values.	1
I5	Receipt of records with same uuid and different values	The record is overwritten with values that belongs to another session.	4
I6	Receipt of a label during the production mode	The label is inserted into the raw session object and sent to the Preparation System (solved by the Preparation System)	1
I7	Wrong record content	Record discarded (validation fails)	4
I8	Receipt of records related to different sessions	The sessions are discarded because wrongly considered as incomplete. The error is not propagated	5
I9	Receipt of records related to the same session in random order	The records are synchronized during the building of the raw session in the data store	1
<b>TOTAL</b>			25

## Preparation System (Daka)

ID	Input Error	Consequence	Score
P1	Wrong raw session structure	Validation failure	4
I2.1	Missing time series samples in the time_series field (in range)	The time series is recovered by interpolation	1
I2.2	Missing time series samples in the time_series field (out range)	The raw session is discarded because the interpolation is not feasible	4
I5	Receipt of raw session with wrong values	Not detected, the prepared session with incorrect features is sent to Segregation System (development phase) or Production System (production phase)	5
I6	Receipt of a label during the production mode	The label is not used	1
<b>TOTAL</b>		15	

## Segregation System (Pantè)

Input error	Consequence	Score
Wrong prepared session structure	Validation failure	3
Receipt of prepared session with wrong parameters (caused by missing values in time series)	Not detected and sent to development system in the next dataset	5
Total		8

## Development System (Bataloni)

<b>Input error case</b>	<b>Consequence</b>	<b>Score</b>
System configuration file	System shutdown	3
Classifier configuration file	System shutdown	3
Message Manager configuration file	System shutdown	3
Wrong request body from Segregation System	Return 400	1
Request on wrong endpoint	Return 404	1
<b>TOTAL</b>		<b>11</b>

## Production System (Bataloni)

<b>Input error case</b>	<b>Consequence</b>	<b>Score</b>
System configuration file	System shutdown	3
Message Manager configuration file	System shutdown	3
Wrong request body from Development System	Return 400	1
Wrong request body from Evaluation System	Return 400	1
Request on wrong endpoint	Return 404	1
<b>TOTAL</b>		<b>11</b>

## Evaluation System (Bidani)

Input error	Consequence	Score
Wrong prepared session structure	Validation fails and label not saved, server restarts receival process	2
Receival of non-unique uuid with different or the same label values	The error is logged and mitigated, server restarts receival process	3
Missing session label from the part of expert or classifier	The error won't be detected as join in the database won't find correct record	5
Delayed receival of label from either endpoints	A report generation doesn't occur unless a threshold of labels is reached	2
Receival of wrong session label generated by classifier	Error cannot be detected	5
Total		17

# Testing (Non-Automation)

## Ingestion System (Daka)

Configure Ingestion System

Use case steps	Cognitive cost (1-6)	Salary cost (normalized)	Occurrence %	Subtotal
1. The Data Administrator edits the “ip” and “port” and “db” parameters in the configuration file.	1	1.4	100	1.4
2. The Data Administrator edits the “production_window” and “evaluation_window” parameters in the configuration file.	2	1.4	100	2.8
3. The Data Administrator edits the “missing_sample_threshold” parameter in the configuration file.	3	1.4	100	4.2
4. The Data Administrator starts the Ingestion System.	1	1.4	100	1.4
<b>TOTAL</b>			9.8	

## Preparation System (Daka)

### Configure Preparation System

Use case steps	Cognitive cost (1-6)	Salary cost (normalized)	Occurrence %	Subtotal
1. The Data Administrator edits the “ip” and “port” parameters in the configuration file.	1	1.4	100	1.4
2. The Data Administrator edits the “min_value” and “max_value” parameters in the configuration file.	2	1.4	100	2.8
3. The Data Administrator edits the “features” parameter in the configuration file.	2	1.4	100	2.8
4. The Data Administrator starts the Preparation System.	1	1.4	100	1.4
<b>TOTAL</b>				<b>8.4</b>

## Segregation System (Pantè)

Configure Segregation System (System Administrator)

Use case steps	Cognitive cost (1-6)	Normalized salary cost	Occurrence %	Subtotal
The system administrator sets the stage phase	3	1.4	1	4,2
The system administrator sets the ip and port of the segregation, development and preparation systems	3	1.4	1	4,2
The system administrator sets the “db_name”	3	1.4	1	4,2
The system administrator sets the “max_sessions” upper bound	4	1.4	1	5,6
The system administrator sets the splitting parameters	4	1.4	1	5,6
The system administrator runs the system	1	1.4	1	5,6
Total				29,4

Check data balancing (Data Analyst)

Use case steps	Cognitive cost (1-6)	Normalized salary cost	Occurrence %	Subtotal
Analyze the column diagram with an image viewer	4	1.4	1	5,6
Analyze the balancing report with a text editor	4	1.4	1	5,6
Writes “balanced” in the CLI	1	1.4	0.7	0,98
Writes “not balanced” in the CLI	1	1.4	0.3	0,42
Total				12,6

## Check input coverage (Data Analyst)

Use case steps	Cognitive cost (1-6)	Normalized salary cost	Occurrence %	Subtotal
Analyze the radar diagram with an image viewer	4	1.4	1	5,6
Analyze the coverage report with a text editor	4	1.4	1	5,6
Writes "ok" in the CLI	1	1.4	0.7	0,98
Writes "not ok" in the CLI	1	1.4	0.3	0,42
Total				12,6

## Development System (Bataloni)

### Configure system

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrence	Subtotal
1. The ML Engineer edit the production system “ip” and “port” in the configuration file.	2	2	1	4
2. The ML Engineer edits “hidden-layer-size-range”, the “hidden-neuron-per-layer-range”, the “overfitting-threshold” and the “validation-tolerance” in the configuration file.	2	2	1	4
3. The ML Engineer edits “starting-mode”, “ongoing-validation” and the “stage” fields in the configuration file.	2	2	1	4
4. The ML starts the Development System.	1	2	1	2
<b>TOTAL</b>				<b>14</b>

Set number of iterations

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrence	Subtotal
1. The ML Engineer inserts the correct number of iterations.	3	2	1	6
2. The ML Engineer presses the 'OK' button.	1	2	1	2
<b>TOTAL</b>				<b>8</b>

Check learning report

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrency	Subtotal
1. The ML Engineer opens the training plot.	3	2	1	6
2. The curve is not too stiff and the ML Engineer set "yes" for the number of iterations	1	2	0.3	0.6
3. The curve is not too stiff and the ML Engineer set "no" for the number of iterations	1	2	0.7	1.4
<b>TOTAL</b>				<b>8</b>

Check validation result

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrence	Subtotal
1. The ML Engineer opens the validation file “best-classifiers.csv” and “best-classifiers.json”.	1	2	1	2
2. The ML Engineer compares the validation error against the training error.	3	2	1	6
3. There aren’t valid classifier and the ML Engineer discards all the classifiers.	2	2	0.2	0.8
4. There’s only one valid classifier and the ML Engineer insert the classifier’s uuid.	2	2	0.1	0.4
5. There are many valid classifier and the ML Engineer select the less complex of them.	2	2	0.7	2.8
<b>TOTAL</b>				<b>12</b>

Check test result

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrence	Subtotal
1. The ML Engineer opens the validation file “test-result.csv” and “test-result.json”.	1	2	1	2
2. The ML Engineer compares the validation error against the test error wrt to the tolerance cap.	3	2	1	6
3. The results are under the tolerance cap and the ML Engineer insert “yes”.	2	2	0.8	3.2
4. The results are under the tolerance cap and the ML Engineer insert “no”.	2	2	0.2	0.8
<b>TOTAL</b>				<b>12</b>

## Evaluation System (Bidani)

Configure Evaluation System:

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrence	Subtotal
1. System Administrator initializes system configuration by setting 'sufficient_labels threshold', the 'maximum number of tolerated errors' and of the' consecutive ones'	2	2	1	4
2. System Administrator sets messages_configuration file by inserting 'ip' and 'port'	1	2	0.4	0.8
3. System Administrator starts the 'Evaluation System'	1	2	1	2
<b>TOTAL</b>				<b>6.8</b>

Evaluate Classifier:

Use case steps	Cognitive cost	Salary cost (normalized)	Occurrence	Subtotal
1. Human Expert analyzes the generated report and checks the accuracy of classifier and error thresholds	4	2	1	8
2. Human Expert accepts network and passes it to production	2	2	0.6	2.4
3. Human Expert refuses network and requests new configuration	2	2	0.4	1.6
<b>TOTAL</b>				<b>12</b>

# Testing (Non-Interoperability)

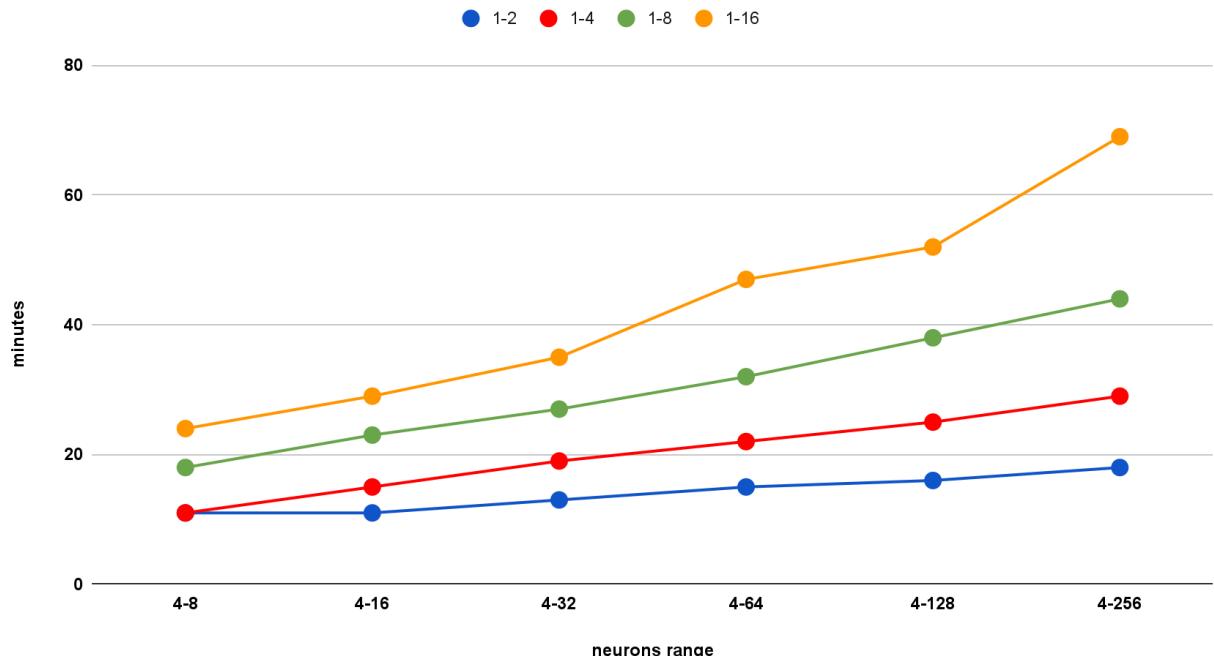
Whole System (Pantè, Bataloni)

Features	Ingestion	Preparation	Segregation	Development	Production	Evaluate	Score
Not evaluation phase (ingestion system)	y	y	n	y	n	-	2
Not in development phase (preparation system)	y	y	-	-	n	n	2
Evaluation phase (ingestion system)	y	-	-	-	-	n	1
Total							5

# Testing (Non-Responsiveness)

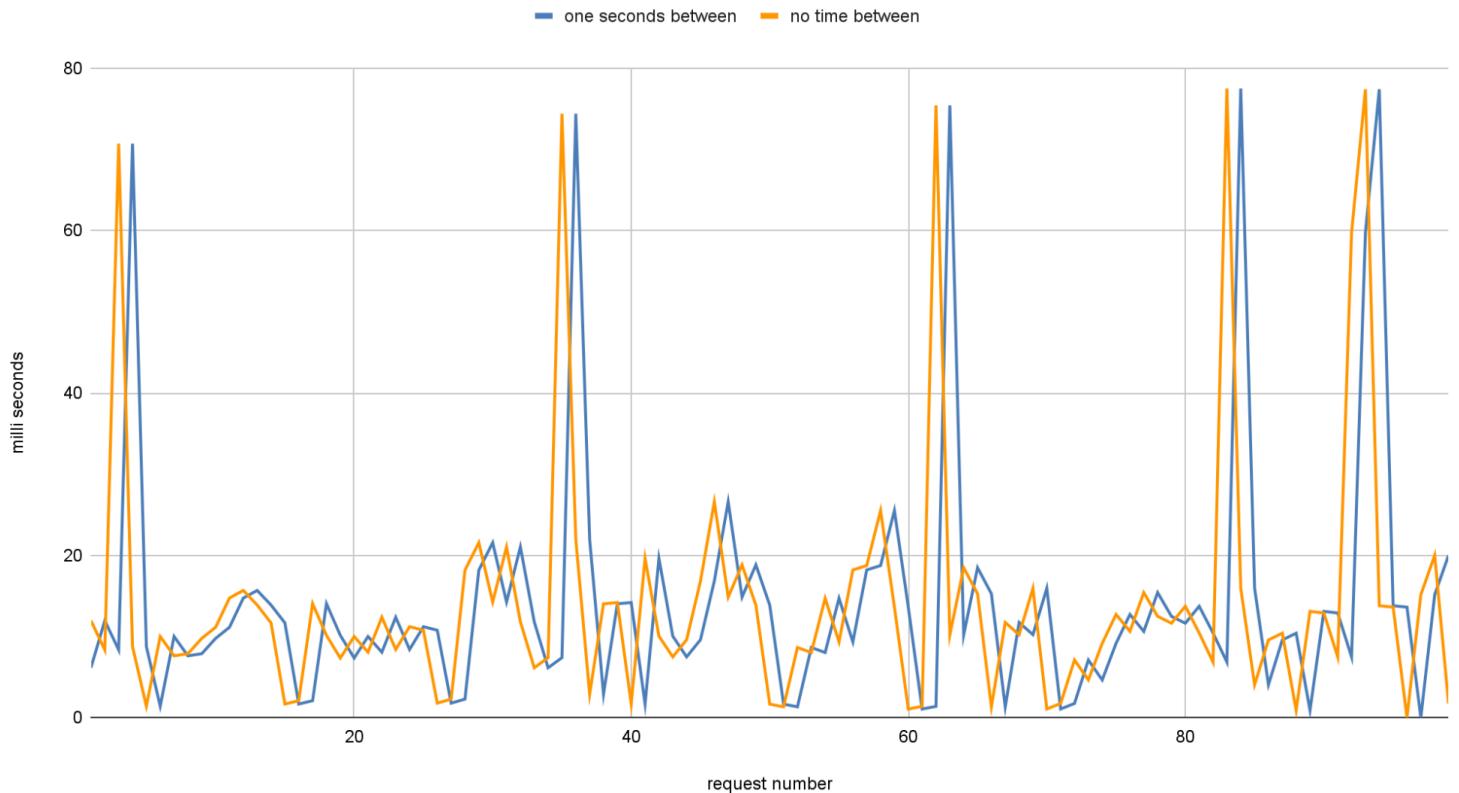
Development Phase (Pantè, Bataloni, Daka)

Development time



## Production Phase (Pantè, Daka, Bataloni)

Time between prepared session and label



Time between ingestion system and label elaborated

